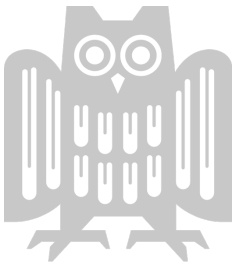
Perceptually driven methods for improved gaze-contingent rendering

Dissertation zur Erlangung des Grades der
Doktorin der Ingenieurwissenschaften (Dr.-Ing.) der
Fakultät für Mathematik und Informatik der
Universität des Saarlandes

Vorgelegt von
Elena Arabadzhiyska-Koleva



Saarbrücken, 2023

Dean:	Prof. Dr. Jürgen Steimle
Date:	June 21st, 2023
Chair:	Prof. Dr. Philipp Slusallek
Reviewers:	Prof. Dr. Piotr Didyk
	Prof. Dr. Hans-Peter Seidel
	Prof. Dr. Belen Masia
Academic Assistant:	Dr. Thomas Leimkühler

Abstract

Computer graphics is responsible for the creation of beautiful and realistic content. However, visually pleasing results often come at an immense computational cost, especially for new display devices such as virtual reality headsets. A promising solution to overcome this problem is to use foveated rendering, which exploits the limitations of the human visual system with the help of eye trackers. In particular, visual acuity is not uniform across the visual field but it is rather focused in its center and it is rapidly declining towards the periphery. Foveated rendering takes advantage of this feature by displaying high-quality content only at the gaze location, gradually decreasing it towards the periphery. While this method is effective, it is subject to some limitations. An example of such limitation is the system latency, which becomes noticeable during rapid eye movements when the central vision is exposed to low-resolution content, reserved only for the peripheral vision. Another example is the prediction of the allowed quality degradation, which is based solely on the visual eccentricity; however, the loss of the peripheral acuity is more complex and it relies on the image content as well.

This thesis addresses these limitations by designing new, perceptually-driven methods for gaze-contingent rendering. The first part introduces a new model for saccade landing position prediction to combat system latency during rapid eye movements. This method extrapolates the gaze information from delayed eye-tracking samples and predicts the saccade's landing position. The new gaze estimate is then used in the rendering pipeline in order to forestall the system latency. The model is further refined by considering the idiosyncratic characteristics of the saccades. The second part of this thesis introduces a new luminance-contrast-aware foveated rendering technique, which models the allowed peripheral quality degradation as a function of both visual eccentricity and local luminance contrast. The advantage of this model lies in its prediction of the perceived quality loss due to foveated rendering without full-resolution reference. As a consequence, it can be applied to foveated rendering to achieve better computational savings.

Zusammenfassung

Die Computergrafik ist für die Erstellung schöner und realistischer Inhalte verantwortlich. Visuell ansprechende Ergebnisse sind jedoch oft mit einem immensen Verarbeitungsaufwand verbunden. Eine Lösung zur Überwindung dieses Problems ist das foveale Rendern, bei dem die Einschränkungen des menschlichen Sehsystems mit Hilfe von Eye-Trackern ausgenutzt werden. Von besonderer Bedeutung ist, dass sich die Sehschärfe nicht gleichmäßig über das gesamte Gesichtsfeld verteilt, sondern sich eher im Zentrum konzentriert. Das foveale Rendern macht sich diese Eigenschaft zunutze, indem es hochwertige Inhalte nur im Zentrum des Blickfeldes anzeigt und die Qualität zur Peripherie hin allmählich abnehmen lässt. Diese Methode ist zwar effektiv, unterliegt aber einigen Einschränkungen. Ein Beispiel ist die Latenz des Systems, die sich bei schnellen Augenbewegungen bemerkbar macht, wenn das zentrale Blickfeld auf niedrig aufgelöste Inhalte fällt, die nur für das periphere Blickfeld gedacht waren. Ein weiteres Beispiel ist die Abschätzung der tolerierbaren Qualitätsminderung, wenn sie ausschließlich auf der visuellen Exzentrizität basiert; der Verlust der peripheren Sehschärfe ist jedoch komplexer und hängt auch vom Bildinhalt ab.

In dieser Arbeit werden diese Einschränkungen durch die Entwicklung wahrnehmungsgesteuerter Methoden für blickabhängiges Rendern angegangen. Zuerst wird ein Modell zur Vorhersage der Landeposition von Sakkaden eingeführt, um die Systemlatenz zu verringern. Diese Methode extrapoliert die Blickinformationen aus verzögerten Eye-Samples und prognostiziert die Landeposition der Sakkade. Die neue Blickschätzung wird dann im Rendern verwendet. Das Modell wird weiter verfeinert, indem die idiosynkratischen Eigenschaften der Sakkaden berücksichtigt werden. Anschließend wird ein den Luminanzkontrast berücksichtigendes foveales Renderverfahren eingeführt, das die tolerierbare periphere Qualitätsminderung als Funktion sowohl der visuellen Exzentrizität als auch des lokalen Luminanzkontrastes modelliert. Der Vorteil dieses Modells liegt in der Vorhersage des wahrgenommenen Qualitätsverlustes durch foveales Rendern, ohne dass eine Referenz in voller Auflösung benötigt wird. So kann es beim Rendern angewendet werden, um eine bessere Rechenleistung zu erzielen.

Acknowledgements

First and foremost I wish to express my deepest gratitude towards Piotr Didyk. His decision to give me a chance to pursue a doctorate, also his unwavering support and brilliant ideas throughout the years enabled the creation of this dissertation. His excellent supervision pushed me to attend to every detail and helped me to become a researcher. I also want to thank Cara Tursun for her pivotal involvement in every project included in this work. To Karol Myszkowski, I want to thank him for his knowledgeable input and support. To Hans-Peter Seidel, for granting me permission to be part of the Max-Planck Institute for Informatics and thus to finish my thesis. I am also grateful to the amazing people I met throughout my journey, who made it so much more pleasant: Michal Piovarči, Dushyant Mehta, Thomas Leimkühler, and Jozef Hladký among others.

Last but not least I want to thank my family, especially my husband Pavel and my son Philip, for their love, patience, and sacrifices. Only with their support, I could reach this milestone in my life.

Contents

Contents **ix**

List of Figures **xiii**

List of Tables **xxiii**

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions 4
1.3 Outline 5

2 Background **7**
2.1 Human visual system 7
2.1.1 The eye and its retina 8
2.1.2 Visual cortex 11
2.2 Eye movements 12
2.2.1 Smooth pursuit movement 12
2.2.2 Vestibulo-ocular movements 13
2.2.3 Vergence movements 13
2.2.4 Saccades 14
2.3 Contrast perception 17
2.3.1 Peripheral vision 18
2.3.2 Blur sensitivity 19

3 Related Work **21**
3.1 Saccade Landing Position Prediction 21
3.2 Foveated Rendering 23
3.2.1 Traditional techniques 24
3.2.2 Image metrics 25
3.2.3 Content-dependent techniques 25

4 Saccade Landing Position Prediction for Gaze-Contingent Rendering 27
4.1 Overview . . . 29
4.1.1 Measurements . . . 29
4.1.2 Data Processing . . . 30
4.1.3 Model . . . 33
4.2 Validation . . . 40
4.2.1 Guided-Viewing Experiment . . . 40
4.2.2 Free-Viewing Experiment . . . 41
4.3 Discussion and Future Work . . . 45
4.4 Conclusion . . . 47

5 Practical Saccade Prediction for Head-Mounted Displays: Towards a Comprehensive Model 49
5.1 Introduction . . . 49
5.2 Overview . . . 50
5.3 Experiment design . . . 51
5.3.1 Stimuli . . . 51
5.3.2 Task . . . 52
5.3.3 Hardware . . . 53
5.4 Analysis of experimental data . . . 54
5.4.1 Saccade profiles extraction . . . 54
5.4.2 Dissimilarity measure for saccade displacement profiles . . . 56
5.4.3 Discussion . . . 57
5.5 Method for tuning saccade prediction models . . . 61
5.5.1 Shearing saccade profiles . . . 63
5.5.2 Computation of shearing transformation between saccadic profiles . . . 63
5.5.3 Application . . . 64
5.5.4 Results . . . 67
5.6 Conclusion . . . 73

6 Luminance-Contrast-Aware Foveated Rendering 75
6.1 Introduction . . . 75
6.2 Overview . . . 77
6.3 Computational Model . . . 78
6.3.1 Perceptual Contrast Measure . . . 78
6.3.2 Estimation of Resolution Reduction . . . 81
6.4 Calibration . . . 84
6.5 Implementation . . . 93

6.6 Validation 94
6.6.1 Visual Evaluation 95
6.6.2 Foveated vs. Non-Foveated Rendering 95
6.6.3 Local vs. Global Adaptation 97
6.6.4 Further validation 98
6.7 Limitations and Future Work 98
6.8 Conclusion 101

7 Conclusion **103**

Bibliography **107**

Figures

1.1 The HTC VIVE Pro Eye Headset - a look from outside (left), and from the inside (middle), showing the infrared LEDs used for eye tracking. The picture on the right presents an example of the traditional foveated rendering technique. [Guenter et al., 2012] . . . 2

1.2 A simplified example of a scanpath where images a) to d) represent fixations between rapid eye movements (saccades). Image f) shows how the information accumulated through the fixations is integrated in our brains. 3

2.1 High-level model of the human visual system. 7

2.2 Anatomy of the eye. Once the light enters the eye though the cornea and the lens, it has to travel though several transparent layers of cells until it reaches the photoreceptors at the back of the retina. The fovea is the region with highest concentrations of cones, responsible for the daylight and color vision. The blind sport is devoid of any photoreceptors and does not contribute to the vision. 8

2.3 Left: The distributions of both cones (blue line) and rods (red line) photoreceptors. The highest concentration of cones is observed in the middle of the fovea and their density rapidly declines towards the periphery. In comparison, there are no rods in the foveal center - their concentration peaks at around 20° and then gently declines with the increase of the periphery. Plot adapted from Weier et al. [2017]. Right: An illustration of what fraction of the monocular field of view is the central foveal vision (right eye). 9

2.4 Left: Illustration of the mapping between foveal area in the retina and the cortical length in the visual cortex. (adapted from Weier et al. [2017]). Right: The linear expansion of the visual degrees processed by a single millimeter of cortical distance with the increase of the eccentricity (adapted from Schor [2011]). 11

2.5 Left: SPEM performed at three different velocities to follow a slowly moving target in three separate occasions. There is a catch-up saccade in the beginning allowing the eyes to align with the target before the tracking continues. Right: Saccade performed after the sudden relocation of the OOI. Plots are adapted from Purves et al. [2001] 13

2.6 Development of the saccadic suppression for three ranges of stimulus positions: left periphery, center, and right periphery. The time is relative to the saccade onset. Plot is adapted from Knöll et al. [2011]. 16

2.7 Left: An illustration of the contrast sensitivity function. In the area below the white line the observer is able to distinguish the sinusoidal pattern, whereas the area above the line appears completely gray. Note, that the actual position of the function depends on the retinal size of the Campbell-Robson chart and on the luminance of the media used to present it. Right: Hypothetical contrast threshold as a function of eccentricity for various spatial frequencies. Plot adapted from Peli et al. [1991]. 18

3.1 Left: Saccade displacement profile (blue) and the gaze estimations used by the system. Notice the latency. Right: Two possible ways to utilize saccade position prediction: either by making prediction for the next frame (tracking) or make a direct prediction for the landing position. Methods for both approaches are introduced by Han et al. [2013a] from where the two plots were adapted. 22

3.2 Left: Standard foveated rendering using three layers of a single frame, rendered in various resolutions and sizes. Image from Guenter et al. [2012]. Middle: Perceptually-enhanced foveated rendering. Image from Patney et al. [2016]. Right: Perceptually-adapted sampling pattern for foveated sparse shading. Image from Stengel et al. [2016]. 24

4.1 Standard gaze-contingent rendering (top row) updates the image according to the current gaze prediction. Due to the system latency, during a saccade, there is a significant mismatch between the rendering and the actual gaze position (b, c). The method moves the foveated region to the actual gaze position only after a delay equal to the system latency (d). Our technique (bottom row) predicts the ending position of the saccade at its early stage and updates the image according to the new prediction as soon as it is available (b). Due to the saccadic suppression the user cannot observe the image manipulations during the saccade (b). When the saccade ends and the suppression is deactivated (c), the observer sees the correct image at the new gaze position with our method. 27

4.2 Our measurement setup and a sample subset of eye-tracking data from one participant. The white circles visualize saccade targets shown during the experiment, whereas the blue traces correspond to eye-tracker samples. For simplicity, only 12 targets are visualized here. Throughout the actual experiment, 300 consecutive targets were shown to each observer. 30

4.3 Visualization of our data processing for one sample saccade. Top: Gaze samples for a saccade from the bottom-right corner to the top-left corner of the screen. Bottom: The gaze velocity and displacements. Samples corresponding to the anchor, detection and the end points are indicated in green, cyan and yellow colors, respectively. Captured with a Tobii TX300 eye tracker at 300 Hz sampling frequency. 32

4.4 Displacement profiles of one participant for different saccade amplitudes are given in (a). The corresponding prediction surface is shown from two different viewing angles in (b) and (c) with the saccade amplitude in the z-axis. Standard deviation of the prediction surface across all participants is given in (d). For simplicity, only 5 individual saccades are shown in (a), (b) and (c). We collected more than 300 saccades from each participant. 34

4.5 Comparison of different prediction modeling methods. Left: the mean absolute error as a function of normalized saccade time. Right: the standard deviation of the mean absolute error. 36

4.6 Gaze samples (yellow), landing position predictions (green) and corresponding prediction intervals (blue) for a sample saccade. The beginning of the saccade (orange), gaze samples and predictions are shown as the square, diamond and circular shapes, respectively. Color saturation level of the points indicates the time when each sample is observed and each prediction is made (more saturated color indicates more recent sample and prediction). Arrows connect gaze samples with our model's corresponding predictions for the landing position. 37

4.7 Left: The mean absolute error of personalized models is compared with that of the average models using interpolation and polynomial fitting. Right: The amount of improvement in the mean absolute errors when personalized models are used instead of the average model for each participant. 38

4.8 Left: Four sample synthetic stimuli, each consisting of four Landolt's C shapes. The size of each stimulus was 1.2°. During the experiment, the orientations and the index of the blurry shape was randomly chosen. The blur was removed when the foveated region moved to the position of the stimulus. Right: Results of the user experiment. A value closer to 0.25 (expected success ratio with purely random choice) is considered more successful at hiding potential artifacts due to the transition from non-foveated to foveated rendering at the positio=n of the stimulus. The error bars correspond to the standard deviation across participants. . . 42

4.9 Left: Insets of the images used in the free-viewing experiment. The size of each image is 2560 × 1440 pixels. Right: Results of the free-viewing experiment. The error bars correspond to the standard deviation across participants. 43

5.1 The figure presents the main stages of each trial of our experiment. In the initial phase, we had either static initialization (left), where the initial gaze was shown as a static target in the center of the screen, or dynamic initialization (middle), where the initial gaze was moving to stimulate a smooth pursuit eye movement. After 1–2 seconds of the initial phase, the sphere was displaced to stimulate a saccade. Some trials of the experiment included a change in depth to stimulate vergence eye movement as shown on the right. 51

5.2 The images of the stimuli as shown to the participants of the experiment on the VR display. 52

5.3 Mean displacements ($\overline{d_l^k}$) and standard deviations (σ_l^k) that we used in Equation 5.2 are shown for hypothetical mean saccade profiles. Mean displacement profiles of three categories; namely, $\overline{S^0}$, $\overline{S^1}$, and $\overline{S^2}$, are represented by green, pink, and brown solid lines, respectively, whereas whiskers and dotted lines visualize the standard deviation of saccade displacements from the corresponding category. 57

5.4 Effects of different factors on the saccade mean displacement profiles computed from our experiment data. The solid lines represent mean profiles for each category, while the dashed lines visualize corresponding standard deviations. The bar plots show the values of our profile dissimilarity measure (Equation 5.2) for different factors (Section 5.4.1). The bars representing the AMPLITUDE factor are provided as a reference baseline for the minimum value of similarity to observe a significant effect (please refer to Section 5.4.1 for details). 58

5.5 The relation between the vergence change during the saccade and the peak velocity. The blue points correspond to individual saccades, while red curves are the quadratic line fits showing the overall trend. Negative values indicate saccades that move closer to the observer, whereas positive values indicate saccades moving further away. 59

5.6 Two examples of shearing original mean saccade profiles to match different targets. The target on the left represents a category with slower saccades than the original. The target on the right represents a category with faster saccades. 64

5.8 Two examples of shearing saccade profiles, recovered from a model, to match the mean target profiles. The target on the left represents a category with faster saccades than the original recovered profile. The target on the right represents a category with slower saccades. 66

5.7 The linear function $f(\alpha)$ (red) is fitted to the shearing factors λ_{α_i} (blue). 66

5.9 The figure presents the performance of differently derived models for horizontal and vertical saccades. The left-most plot presents aggregate mean errors for predictions made for the entire saccade duration (height of the bar) and for the second half of the duration (light segment). The two other plots present the error as a function of duration of the saccade, i.e., at which stage of the saccade the predictions was performed. While the *customized model* performs best, the *sheared model*, which requires significantly lower number of saccades for training, performs better than the *average model*, which does not account for the orientation of the saccade. 69

5.10 Comparison of the mean saccade profiles from the full dataset (pink) and from the vertical subset (orange) with a recovered one from the *average model* (green). The difference between the two mean profiles indicates slower performance of the vertical saccades with respect to the rest but the sample profile recovered from the average model does not reflect it. 70

5.11 Performance comparison for different models as a function of the number of saccades used for their computation. While the plot on the left shows the average error of the prediction for the full length of the saccade, the center plot shows the error for the prediction during the second half of the duration. The solid lines are the means computed using bootstrapping with 20 repetitions, the dotted lines are the corresponding standard deviations. The plot on the right compares the average error of the prediction at any point during the saccade when using 10 and 200 saccades for training the models. 71

5.12 The figure presents the performance of different models for each user. The bar plot on the left shows aggregate mean absolute errors for the saccade amplitude predictions. The height of the bars represents the mean error measured for whole duration of the saccade while the segments shaded with lighter colors represent the mean error measured in the second half of the saccade duration. The line plot on the right shows the mean absolute error as a function of point in time when the prediction was made during the saccade. The *customized model* gives the best performance, followed by *model shear* and then the *average model* (please see the text for details). *Model shear* mostly has a good prediction performance for the users, for whom the *customized model* also performs well. 72

6.1 Current foveated rendering techniques (left) use a fixed quality decay for peripheral vision. While this can be a conservative solution, it does not provide a full computational benefit. Our technique (right) performs content-adaptive foveation and relaxes the quality requirements for content for which the sensitivity of the human visual system at large eccentricities degrades faster. 75

6.2 The same foveation exhibits different visibility depending on the underlying texture. In this image, the foveation was optimized such that it is invisible for the photograph (left part). At the same time, however, it can be easily detected on the text texture (right part). 76

6.3 Overview of our method. Our predictor takes patches, retinal eccentricity, observer distance and display parameters such as the resolution, gamma, peak luminance, physical width and height as inputs and predicts the required spatial rendering bandwidth expressed as the standard deviation of a low-pass Gaussian filter. The map is generated using our method and the output is enhanced for visibility. Image by Pxhere. 79

6.4 This figure shows a flowchart of our model for computing the perceptual contrast measure. The input parameters which are optimized during the calibration are shown in bold. 80

6.5 Our dataset for the calibration of our predictor. We include patches with different luminance and texture patterns from a dataset of natural and synthetic images [Cimpoi et al., 2014]. Note that patches 1-12 contain reduced-contrast versions of the same content to cover a wider range of contrasts during the calibration phase. 86

6.6 A sample stimulus used for data collection and calibration. The zoomed region shows how the input patch is tiled prior to Gaussian filtering. For different values of foveal region radius r and rate of quality drop-off k, the participants are asked to compare foveated (shown here) and non-foveated (without Gaussian blur) versions in a 2AFC experiment. 87

6.7 Box plot of ground truth $\sigma_s^{(i)}$ obtained from our experiment. This plot shows how content influences the tolerable amount of foveation with respect to eccentricity. Red lines represent the median while boxes show the range between 25th and 75th percentile of the data. Whiskers extend to the whole range. The patches which have the minimum, the median and the maximum $\sigma_s^{(i)}$ are shown on the plot for 30° eccentricity. 88

6.8 These plots show how $\sigma_s^{(i)}$ (y-axis) changes with respect to eccentricity (x-axis) for each patch. The lines represent the mean (blue) and the standard deviation (red) for the given patch across all participants. 89

6.9 These plots show how $\sigma_s^{(i)}$ (y-axis) changes with respect to eccentricity (x-axis) for each patch. The lines represent the mean (blue) and the standard deviation (red) for the given patch across all participants. 90

6.10 We used a Campbell-Robson chart (top left) as a test input for our predictor. The predictions of σ_s from our model are given for different visual eccentricities. The eccentricities are indicated at the top-right corner of each map. Our model successfully predicts a higher σ_s (corresponding to a lower rendering resolution) for the spatial frequencies that are imperceptible by the HVS as the visual eccentricity increases and the contrast declines. (Please note that the Campbell-Robson chart is prone to aliasing when viewed or printed in low resolution. Please refer to the electronic copy of this document for a correct illustration.) 92

6.11 The outputs from our model for the images used in our validation experiments. The gaze position is fixed at the center for all images. The heatmaps (right) show the predicted standard deviations ($\hat{\sigma}_s$) of a low-pass Gaussian filter which results in a contrast loss of 1 JND when applied on the input (left). $\hat{\sigma}_s = 0$ represents a requirement for rendering in the native display resolution whereas larger values represent rendering in a lower resolution. Average value of each $\hat{\sigma}_s$ map is shown in the top-left corner. These results show how the effect of content is captured by our method to adapt foveation strength for rendering. Image 1 is by Manu Jarvinen. 96

6.12 Detection rates of the participants for our method and non-foveated rendering. x-axis represents different multipliers that we use for changing the average σ_s prediction to test the effect of different rendering budgets on the preferences of participants. The actual prediction of our method corresponds to the multiplier value of 1 and increasing values on the x-axis represent more limited rendering budgets. The trend in the detection rate shows that the participants actually detect the foveation and the detection rate for the actual rendering is smaller than 0.75 for all platforms when the predictions are not scaled. The error bars represent standard error. 97
6.13 The result of our subjective experiments where the participants compared our method with globally adaptive foveated rendering, which does not take local distribution of contrast into account. The error bars represent standard error. 99
6.14 Average foveal region size preferences of the participants for each image for globally adaptive foveation. We observe a high variability between the images and we attribute this to the role of content in different images. This data shows that a traditional foveated rendering with a fixed foveal region size would not provide the optimal perceived quality. The error bars represent standard error. 100

Tables

5.1 The factors that we consider when analyzing saccades and the categories in which we classify them according to each individual factor. 55
5.2 Short descriptions of the four models that we compare in Section 5.5.4. Each model is created following the procedure described in Chapter 4, either using our full dataset or a subset of it that includes a single category of saccades (Table 5.1). For *data shear* we modify the dataset before creating the model and for *model shear* we first create the model and then modify it to match a specific subset. 68

6.1 Optimal parameter values obtained during calibration and corresponding cross-validation errors. 91
6.2 Best parameter values obtained after calibration. The input patches are downscaled by a factor of 1/4. Loss is the training error computed using Equation 6.19. In addition to the loss function, which is a weighted mean absolute error, we also provide the standard unweighted mean absolute error (MAE) for evaluation. 93
6.3 Running times of our implementation. Our predictor is calibrated and validated using 1/4× downsampled inputs (1/16× of the area) in a series of subjective experiments. Here, we show the computational savings obtained by our approach with respect to the predictions from full-resolution inputs. Please note that these values do not include rendering costs. 94

Chapter 1

Introduction

This chapter provides the motivation for designing perceptually-driven models for foveated rendering (Section 1.1), summarizes a list of our contributions and publications (Section 1.2), and gives an outline of this dissertation (Section 1.3).

1.1 Motivation

One of the major goals of computer graphics is to create beautiful, realistic, and rich in details visual content. To achieve this goal complex algorithms for image synthesis are required, as well as an interactive and immersive environment. These requirements come with an immense demand for computational power, especially for the novel devices such as head-mounted displays (HMDs) that provide virtual, augmented, and mixed reality to the viewer. Over the past decade, the popularity of the virtual reality technologies has been on a steady increase but the content provided by them is still sub-optimal. While the high resolution, reaching up to 8K per eye, is still insufficient to completely remove visible pixelation, it is already straining the performance of the hardware system. One way to elevate this problem is to introduce alterations and simplifications to the rendering pipeline but, if done without any consideration, this approach inevitably diminishes the quality and undermines the realism of the synthesized image. Therefore, we should seek a way to decrease the enormous demand for computational power without negatively affecting the user experience. Guenter et al. [2012] propose a simple yet very efficient solution to this problem: Exploit the human visual system by using foveated rendering.

The human visual system (HVS) involves our eyes, which are our light detectors, and our brain, which processes the information that comes from our eyes to form a mental image of the world that surrounds us. This system, however

complex, is far from perfect. One of its limitations is that the visual acuity is not uniform across the entire visual field. Despite the wide field of view of the human eye, only a fraction - a region of around 5° called fovea - is dedicated to high definition vision. The visual acuity gradually declines towards the periphery. Foveated rendering [Guenter et al., 2012] exploits this limitation by providing high-resolution content only to the central vision and decreasing the quality for the peripheral vision thus using only a portion of the resources needed to render an image in its entirety. Key enablers of this technique are the eye trackers - devices that capture and process images of our eyes to estimate our gaze location on the screen. The prices of the eye trackers have been on the decline for the past years, making the low-end devices affordable for the mass consumer; recent HMD models even come with such equipment integrated into them (Figure 1.1, left and middle). Foveated rendering, however, suffers from its own limitations: first, it is subject to system latency, which arises from the low sampling rate of the commercially-available eye trackers; second, the traditional approach proposed by Guenter et al. [2012] uses radially symmetric quality degradation, which disregards the image content and our perceptual response to it.



Figure 1.1. The HTC VIVE Pro Eye Headset[1]- a look from outside (left), and from the inside[2](middle), showing the infrared LEDs used for eye tracking. The picture on the right presents an example of the traditional foveated rendering technique. [Guenter et al., 2012]

The first limitation, system latency, comes from the time needed for capturing the image of an eye, the rendering, and the displaying of the synthesized image to complete. It becomes a problem for foveated rendering and other gaze-contingent methods when there is a significant mismatch between the actual eye position on the screen and the one used for displaying the current frame. This may lead to visible artifacts such as the perception of a low-quality image when

[1] https://www.vive.com/us/product/vive-pro-eye/overview/
[2] https://www.cnet.com/

the fovea is exposed to low-quality content. The system latency is most pronounced during rapid eye movements called saccades, which are the fastest type of eye movements. Due to the degrading visual acuity towards the periphery, the human eyes are incapable of observing the world around us with a single fixation. To compensate, we naturally and mostly unconsciously make a succession of saccades to capture every detail with our central vision, as illustrated in Figure 1.2 – a) to d). The observed features are accumulated and we are given an impression of a crisp picture of our surroundings (Figure 1.2 – f)). The average time, required for a single saccade to complete, is approximately $20-80ms$. The average latency for the HTC Vive Pro Eye is estimated to be around $80ms$ [Stein et al., 2021] but it can reach a higher number, depending on the complexity of the presented scene. As a result, at the end of the saccade, the distance between the actual gaze position and the one predicted by the system might be significant enough to lead to a degraded user experience.

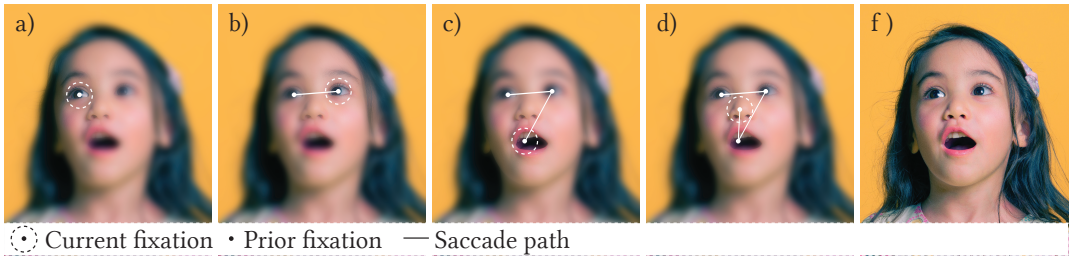


Figure 1.2. A simplified example of a scanpath where images a) to d) represent fixations between rapid eye movements (saccades). Image f) shows how the information accumulated through the fixations is integrated in our brains.

The second limitation of the foveated rendering technique lies in the prediction of how much distortions a viewer can tolerate in the periphery: It is usually expressed as a function of eccentricity but neglects the fact that the sensitivity of the HVS to image distortions also depends on the underlying content - the effect known as visual masking. A relevant observation in this thesis is that the visibility of foveation depends on the underlying luminance contrast, i.e., while a given reduction of spatial resolution becomes objectionable in high-contrast regions, it remains unnoticed for low-contrast regions. There is a significant difference in the tolerable amount of quality degradation depending on the underlying visual content. In Figure 1.1 – right the reader may observe that the sky in the background of the image allows for considerably greater amount of blur in comparison to the building structures in the front. However, in order to guarantee the quality of the user experience, a conservative level of foveation should be

chosen so that it accounts for the most visible peripheral distortions.

1.2 Contributions

The main contributions of this dissertations are as follows:

- **Saccade Landing Position Prediction** In the first work we address the problem of system latency that hampers the utilization of the gaze-contingent rendering. To this end, we suggest a new way of updating images in gaze-contingent rendering during saccades. Instead of rendering according to the current gaze position, our technique predicts where the saccade is likely to end and provides an image for the new fixation location as soon as the prediction is available. While the quality mismatch during the saccade remains unnoticed due to saccadic suppression, a correct image for the new fixation is provided before the fixation is established. This work describes the derivation of a model for predicting saccade landing positions and demonstrates how it can be used in the context of gaze-contingent rendering to reduce the influence of system latency on the perceived quality. The technique is validated in a series of experiments for various combinations of display frame rate and eye-tracker sampling rate.

- **Practical Saccade Prediction for Head-Mounted Displays** Following the previous work, we go beyond existing models for predicting the saccade landing position by investigating additional factors that affect these eye movements and their prediction. More specifically, we focus on dynamic scenarios in VR and AR devices where saccades are combined with vergence movements and smooth pursuit eye motion (SPEM). Consequently, we design and conduct user experiments which measure saccade profiles in such scenarios. Then, we analyze the profiles and compare different factors and their impact on the saccades. Finally, we propose a method for correcting existing models to account for these factors.

- **Luminance-Contrast-Aware Foveated Rendering** In this work we look at the allowed quality degradation used in foveated rendering and investigate ways to improve it by considering factors additional to the eccentricity. To this end, we propose a new luminance-contrast-aware foveated rendering technique which demonstrates that the computational savings of foveated rendering can be significantly improved if local luminance contrast of the image is analyzed. To achieve this, we first study the resolution requirements at different eccentricities as a function of luminance patterns. We

later use this information to derive a low-cost predictor of the foveated rendering parameters. Its main feature is the ability to predict the parameters using only a low-resolution version of the current frame, even though the prediction holds for high-resolution rendering. This property is essential for the estimation of required quality before the foveated image is rendered. We demonstrate that our predictor can efficiently drive the foveated rendering technique and analyze its benefits in a series of user experiments.

List of related publications:

- **Elena Arabadzhiyska**, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, Piotr Didyk, **Saccade Landing Position Prediction for Gaze-Contingent Rendering**, ACM Transactions on Graphics 36(4) (Proceedings SIGGRAPH 2017, Los Angeles, CA, USA)

- Okan Tarhan Tursun, **Elena Arabadzhiyska**, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, Piotr Didyk, **Luminance-Contrast-Aware Foveated Rendering**, ACM Transactions on Graphics 38(4) (Proceedings SIGGRAPH 2019, Los Angeles, CA, USA)

- **Elena Arabadzhiyska**, Cara Tursun, Hans-Peter Seidel, Piotr Didyk, **Practical Saccade Prediction for Head-Mounted Displays: Towards a Comprehensive Model**, ACM Trans. Appl. Percept. Just Accepted (October 2022). https://doi.org/10.1145/3568311

1.3 Outline

This dissertation is divided into 7 chapters. Chapter 2 provides background on the human visual system (HVS) and the dynamics of the rapid eye movements (saccades). Chapter 3 reviews relevant theoretical and practical work on saccade landing position prediction and foveated rendering. Chapters 4, 5, and 6 present the main technical contributions of this thesis and their evaluations: Chapter 4 includes our saccade landing position prediction model, Chapter 5 investigates how different factors influence the saccade landing prediction models, and Chapter 6 introduces our luminance-contrast-aware strategy for foveated rendering. Chapter 7 gives a conclusion to the dissertation and discusses potential future research.

Chapter 2

Background

This chapter aims to share theoretical background, relevant for the gaze-contingent rendering. It focuses on the human visual system (Section 2.1) and the contrast perception (Section 2.3) as they are the most important factors for the design of computationally-efficient methods based on the screen gaze location.

2.1 Human visual system

Sight is probably the most appreciated one of the basic human senses. It is the complex process of transforming light into a comprehensible image, which gives us essential information about the environment that surrounds us: the appearance of different objects - their shape, size and color, their position in space, the direction and the velocity of objects in motion. Entrusted with this task is the human visual system (HVS), which consist of the eyes and parts of the brain (Figure 2.1). Light enters the eye where it is detected by light receptors located in the retina. These photoreceptors transmit the light photons into electrical signals, which are then carried via the optic nerve to the brain. There, at the back of the brain, the visual cortex process these signals into the three-dimensional image that we perceive.

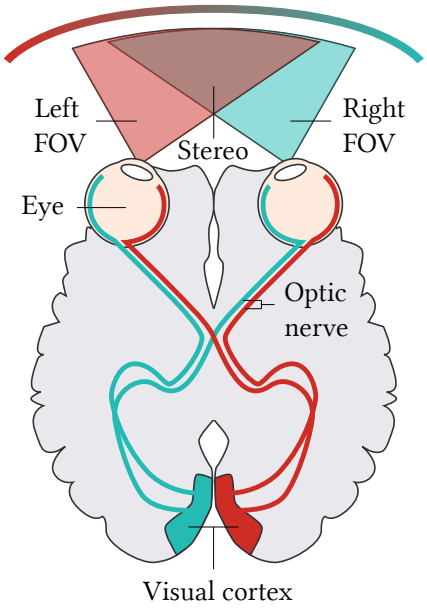


Figure 2.1. High-level model of the human visual system.

In this section we focus on the parts of the HVS that play a major role in the way the visual acuity is distributed across our field of view. Namely, in Section 2.1.1 we discuss the anatomy of the eye and more specifically the retina, which serves as the eye's light sensor and in Section 2.1.2 we discuss the visual cortex, which is where the image formation happens in the brain.

2.1.1 The eye and its retina

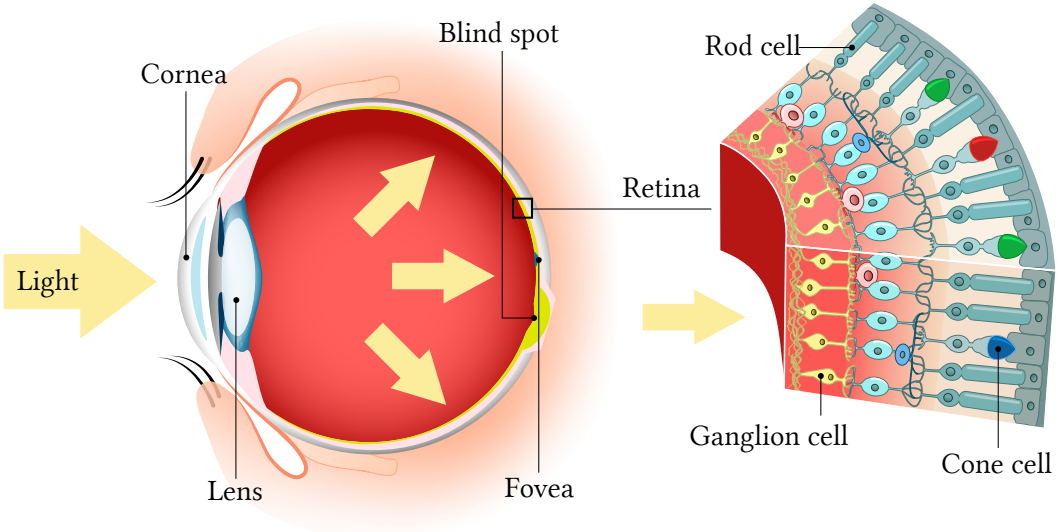


Figure 2.2. Anatomy of the eye. Once the light enters the eye though the cornea and the lens, it has to travel though several transparent layers of cells until it reaches the photoreceptors at the back of the retina. The fovea is the region with highest concentrations of cones, responsible for the daylight and color vision. The blind sport is devoid of any photoreceptors and does not contribute to the vision.

Light enters into the eye through the cornea and with the help of the lens it is focused on the retina. The retina is located in the posterior portion of the eyeball and consist of several layers of cells interconnected by synapses. In Figure 2.2 we can see a schematic illustration of the anatomy of the eye. Densely packed in the outermost layer of the retina are the rod and cone photoreceptors. They serve as the light sensors of our eyes by transforming the light photons into electrical signals, which are passed to the other cell layers until they converge onto the retinal ganglion cells. Through the ganglion cells the visual signals are then sent to the brain via the optic nerve. Directly across the lens lies the fovea. Its

approximate angular subtense is 4°–5°, followed by the parafovea, perifovea and peripheral vision, with each zone containing the entirety of the previous ones [Hendrickson, 2005]. Despite its small relative size, compared to the entire field of view of approximately 160° for a single eye [Spector, 1990], the fovea is the region dedicated for the central human vision where the visual acuity is at its peak. The reason for this being the fact that the fovea contains the highest concentration of the cone photoreceptors, it has also the highest density of ganglion cells and a large portion of the visual cortex is dedicated to processing the information, which is passed on from it. The retina covers the entire posterior portion of the eye, except for the spot where axons pass through to form the head of the optic nerve. This spot is referred to as the blind spot and it is approximately 4° [Gregory and Cavanagh, 2011].

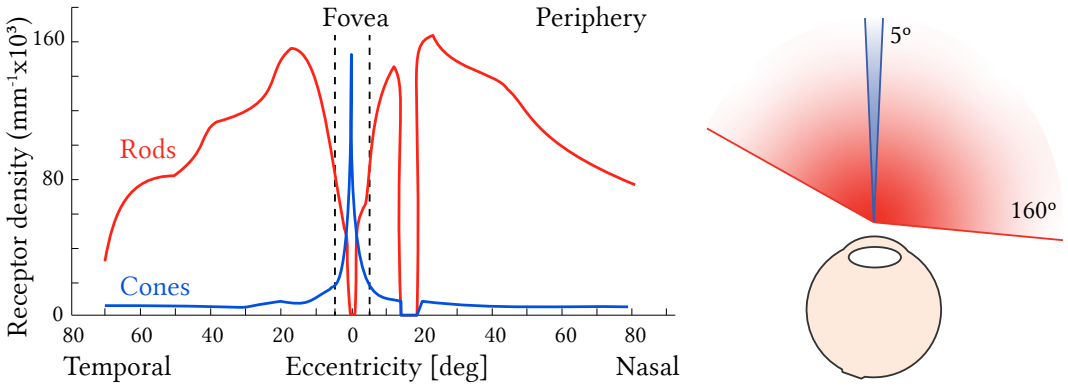


Figure 2.3. Left: The distributions of both cones (blue line) and rods (red line) photoreceptors. The highest concentration of cones is observed in the middle of the fovea and their density rapidly declines towards the periphery. In comparison, there are no rods in the foveal center - their concentration peaks at around 20° and then gently declines with the increase of the periphery. Plot adapted from Weier et al. [2017]. Right: An illustration of what fraction of the monocular field of view is the central foveal vision (right eye).

Cones The photoreceptors responsible for processing the photopic vision are called cones. There are three types of cones, responsible for the perception of different wavelengths [Hendrickson, 2005]: short- (S), medium- (M), and long-wavelength (L), also known as blue, green, and red cones, depending on the color they are sensitive to. There are approximately 6 to 7 million cone cells, comprising only about 5% of the total number of photoreceptors [Mahabadi and Al Khalili, 2022], however, due to their high concentration within the fovea they

are responsible for the high visual acuity of the central vision. The density of the cones within the fovea allows for the eye to resolve a sinusoidal pattern with spatial frequency of 60 cycles per degree, which is matched by the optical filtering occurring in the eye in order to avoid any aliasing effects [Schor, 2011]. Each cone within the fovea converge onto a single retinal ganglion cells, allowing for the densely sampled visual signal to be transported to the brain without filtering. In contrast, the concentration of cones rapidly declines towards the periphery (Figure 2.3) and unlike within the fovea, they no longer converge onto individual ganglion cells.

Rods The photoreceptors responsible for the sctotopic (monochromatic) vision are called rods. Unlike cones, there is only one single type of rods and therefore they all equally respond to different wavelengths without being able to discriminate different colors. The rods are also more sensitive to single light photons and during daytime they are "bleached" and contribute less towards the human vision. The sensitivity of the rods is restored in low-light conditions but it takes approximately 20 minutes for this process to complete. The rods make for 95% of all photoreceptors [Mahabadi and Al Khalili, 2022] in the retina but their distribution differs significantly to the cones - there are no rods in the central region of the fovea, the foveola, after which their concentration rapidly peaks in a ring at approximately 20° before it slowly starts to decline again (Figure 2.3). Since more rods converge onto a single retinal ganglion cell, the rods do not exhibit such high visual acuity as the cones.

Retinal ganglion cells The ganglion cell layer of the retina is much closer to the front anterior of the head, compared to the layer of rods and cones, and it contains the retinal ganglion cells (RGC). The axons of these cells form the optic nerve, through which the image-forming information is passed on to the brain. Although the rods and the cones do not directly synapse onto the RGCs, they converge onto them, therefore the distribution of the RGCs plays an important role in how the visual acuity is formed across the field of view. In comparison to the large number of photoreceptors, there are only 1.07 million ganglion cells on average in the human retina [Hendrickson, 2005]. This leads to unequal coverage of the RGCs in the fovea and in the periphery. In the fovea, the connectivity is one-to-one but in the periphery as many as thousands of photoreceptors converge onto the same RGC resulting in a significant loss of visual acuity. The distribution of the RGCs closely matches the distribution of the cones in the retina, with about 50% of them in close proximity of the fovea and responsible for foveal

signal processing.

2.1.2 Visual cortex

The visual cortex is located in the most posterior region of the brain. Each hemisphere has its own visual cortex responsible for the receiving and processing of visual information incoming from the contralateral eye - the visual cortex in the right side of the brain integrates the signal received from the left eye and the visual cortex in the left side of the brain is responsible for the signal coming from the right eye. The purpose of the visual cortex is to process the information that comes from the eyes and to convert it into three-dimensional image without making a conscious effort to do so. The visual cortex consists of five different areas, named V1 to V5, and each area differs in structure and functionality from the other. V1, also known as the primary visual cortex, is the first area to receive the visual signal and responds to simple stimuli, determining their orientation and direction. This higher-level information is then passed on to the other parts of the visual cortex [Huff et al., 2022].

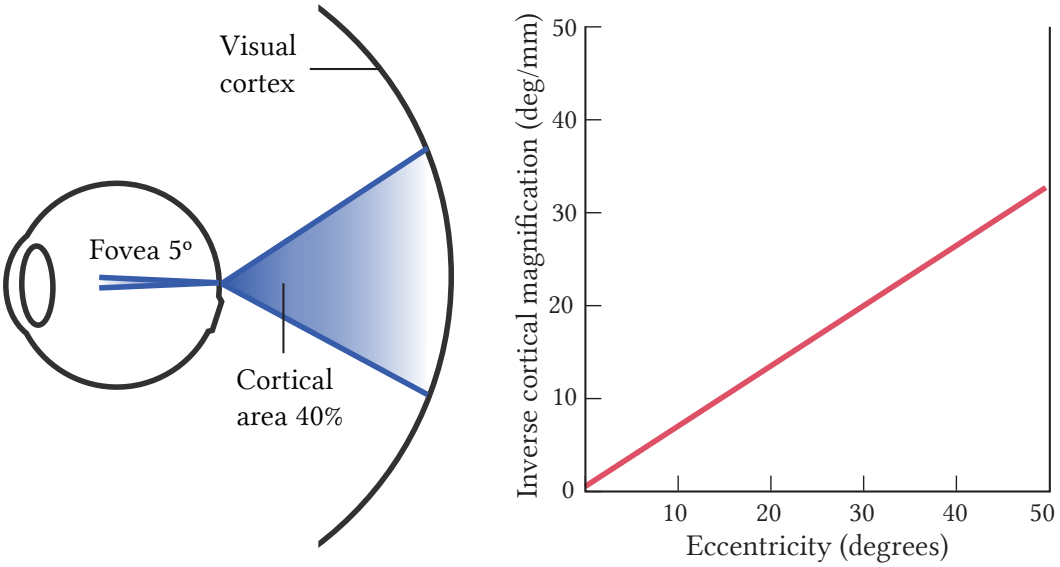


Figure 2.4. Left: Illustration of the mapping between foveal area in the retina and the cortical length in the visual cortex. (adapted from Weier et al. [2017]). Right: The linear expansion of the visual degrees processed by a single millimeter of cortical distance with the increase of the eccentricity (adapted from Schor [2011]).

An important role for the the visual acuity plays the size of the regions of the

primary visual cortex dedicated to processing foveal and peripheral information. The cortical magnification factor (CMF) is the linear extent (usually expressed in mm) of the primary visual cortex devoted to processing a single degree of the visual field. The cortical representation varies across the retina with 40% of it dedicated to processing the fovea [Hendrickson, 2005] (Figure 2.4, left) and the CMF is directly proportional to the human visual acuity [Cowey and Rolls, 1974]. It is estimated that for a single visual degree of the fovea there is about $20mm$ of distance in the cortex representing it, making the CMF for the fovea to be $20mm/°$. In comparison, at eccentricity of 10° the CMF is only $1.5mm/°$ [Schor, 2011]. The inverse function of the CMF (visual degree over cortical distance) is illustrated in Figure 2.4, right.

2.2 Eye movements

In Section 2.1 we discussed how pivotal the narrow foveal vision is to the overall human perception. However, in order to keep objects of interest (OOI) projected onto the fovea, the eyes need to be able to move. There are six extraocular muscles that navigate the rotation of the eye and regulate the adjustment of the lens allowing for the eye to be pointed and focused on the OOI [Weier et al., 2017]. The types of eye movements differ in their function and in this section we will introduce the four basic movements: smooth pursuit, vestibulo-ocular, vergence, and saccades. Due to their particular importance for this thesis, we will focus mostly on saccades and how the human vision is affected before, during, and after these rapid eye movements.

2.2.1 Smooth pursuit movement

Smooth pursuit eye motion (SPEM)occurs when we follow a slowly moving target with our eyes in order to keep a moving target in our fovea. These movements are voluntary in the sense that we can choose whether to follow a target or not. However, it is seldom for SPEMs to be achievable without the presence of a moving target [Purves et al., 2001]. It is estimated that the maximum velocity for SPEM is around $100°/s$ but the accuracy of the motion decreases for velocities larger than $30°/s$ [Weier et al., 2017]. Figure 2.5, left shows an example of how the IOO and the gaze positions change under different conditions for SPEM.

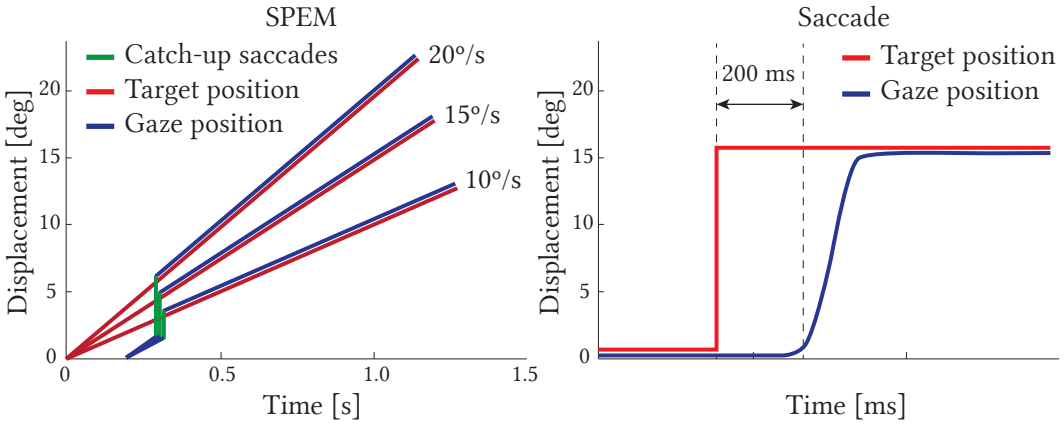


Figure 2.5. Left: SPEM performed at three different velocities to follow a slowly moving target in three separate occasions. There is a catch-up saccade in the beginning allowing the eyes to align with the target before the tracking continues. Right: Saccade performed after the sudden relocation of the OOI. Plots are adapted from Purves et al. [2001]

2.2.2 Vestibulo-ocular movements

Vestibulo-ocular movements are reflex responses designed to keep the OOI within the fovea during head movements. When the head moves in one direction, the eyes automatically respond with a motion in the opposite direction to counter the target offset in the eye created by the head motion. The delay for these reflexes is around $7-15ms$.

2.2.3 Vergence movements

Vergence is the slow ($\approx 0.5-1$s) movement of eyes in the opposite directions when the change in the binocular fixation involves a change in depth. The task of the vergence is to keep the OOI within the fovea for both eyes. They rotate inwards (convergence) when the OOI is approaching and move outwards (divergence) when the OOI is moving further away from the observer. Vergence movements are coupled with accommodation - the compression or relaxation of the lens in order to focus on the relocated target. In the context of VR and AR, where the scene depth is only simulated by showing different images to each eye, vergence occurs without accommodation since the distance to the displayed content does not change. This leads to the vergence-accommodation conflict [Didyk et al., 2011] and may result in unpleasant experience.

2.2.4 Saccades

Saccades are simultaneous movements of both eyes to shift the gaze direction towards the visual stimulus that is away from the point of fixation [Schor, 2011; Leigh and Zee, 2015a]. They are characterized by a rapid acceleration until the maximum velocity is reached, and then a deceleration to full stop, typically followed by corrective small eye movements around the target [Westheimer, 1954]. Figure 2.5, right shows an example of a single saccade profile with amplitude of 15°. performed after the sudden displacement of the OOI.

Saccade characteristics

Pre-programmed behavior Although it is possible to observe saccades up to 100° amplitude, humans perform small saccades more frequently than large ones under natural viewing conditions [Bahill, Adler and Stark, 1975]. Consequently, most of saccades last a brief amount of time ($<$ 70ms), that is approximately equal to the time it takes for visual information to reach the brain's ocular motor mechanisms [Leigh and Zee, 2015a]. Therefore, saccades exhibit a pre-programmed behavior and visual stimuli has negligible effect on a saccade when presented in the last 80-100ms preceding the saccade onset or during a saccade [Young and Stark, 1963; Becker and Jürgens, 1979].

Factors that affect the velocity The velocity of a saccade is affected by multiple factors such as the source and target positions in the visual field, as well as the orientation of their trajectory (e.g., nasal vs temporal) and most notably by the distance between the source and target (i.e., the amplitude of the saccade) [Boghen et al., 1974a]. Initial attempts for studying saccades revealed a relationship of the saccade amplitude to its duration and the peak velocity (a.k.a. the main sequence). It is commonly observed that the duration of the saccades show a nonlinear increase up to approximately 5°, where it starts increasing linearly with the saccadic amplitude [Bahill, Clark and Stark, 1975a; Carpenter, 1988]. Similarly, the peak velocity increases linearly with saccadic amplitude up to 15°–20°, where it reaches a saturation limit at approximately 600°/*s*–800°/*s* [Bahill, Clark and Stark, 1975a]. The initial eye position and the orientation of trajectory also affect their velocity. The saccades which start at the periphery of the orbit and directed towards primary orbital position (centripetal) are on average faster than the saccades performed in the opposite direction (centrifugal) [Pelisson and Prablanc, 1988]. Similarly, the saccades performed in the horizontal direction reach higher peak velocities than those performed in the vertical direc-

tion; however, the difference becomes less significant for older adults [Irving and Lillakas, 2019]. There are some studies which show that the viewed content has an effect on the velocity profiles such that saccades may deviate from a velocity profile that can otherwise be modeled with a compressed exponential model [Costela and Woods, 2019; Han et al., 2013b].

Interactions with vergence If the change in the visual direction also accompanies a change in the depth, saccade and vergence take place simultaneously. In case of such combined saccade and vergence movements, they interact with each other [Ono et al., 1978]. Although saccade takes significantly shorter time ($\approx$50ms) than vergence, a large portion ($40-100\%$) of vergence takes place during the saccade when they are combined [Enright, 1984, 1986]. This shows an effective "mediation" of vergence by saccades. A closer inspection of peak velocities reveals that vergence speeds up while saccade slows down when they are combined [Erkelens et al., 1989; Collewijn et al., 1997; Yang and Kapoula, 2004]. However, the combined eye movement is completed in a shorter duration of time. The speeding up of vergence is observed during both horizontal and vertical saccades [Zee et al., 1992]. However, although combined eye movements are faster than pure vergence, the latency until the onset of eye movements is increased by 18–30ms [Yang et al., 2002]. In addition, the accuracy of saccades is reduced and corrective saccades are required more often when they are combined with vergence [Yang and Kapoula, 2004].

Saccades towards stationary targets during smooth pursuit eye movements The saccades and SPEMs are known to interact with each other. However, experimental data shows that saccades do not add up linearly with SPEMs [Jürgens and Becker, 1975]. On the contrary, for the saccades performed during an ongoing SPEM, the velocity of smooth pursuit is reduced before and after saccades performed in the opposite direction and after saccades performed in the same direction as the pursuit. The decrease is depends on the saccadic amplitude. In order to describe the neural saccade programming process during SPEMs, two types of positional error vectors are defined that may explain the planned amplitude and direction of the saccade; namely, based on *retinal error* and based on *spatial error* [McKenzie and Lisberger, 1986]. When a target is briefly flashed during a SPEM, the eyes stay in the smooth pursuit until the onset of the saccade for approximately 100–200ms. During this brief amount of time the position of the eyes change with respect the initial position when the target was flashed. If the saccades are planned based on the retinal error between source and target

positions, then the neural programming of saccades would take place according to the displacement vector between the initial position and target position without taking into account the displacement of eyes until the onset of the saccade. On the other hand, the saccades programmed according to spatial error would compensate for the displacement of eyes between the initial position and the onset of the saccade. The type of positional error used by the brain to plan saccades determines the accuracy of the saccade. Initial experiments on this matter were contradictive and inconclusive. Some studies showed a correlation of saccades with retinal error [McKenzie and Lisberger, 1986]. Others studies showed a correlation with spatial error if the flash is presented for a longer amount of time [Schlag et al., 1990; Herter and Guitton, 1998]. An explanation for the differences between the results obtained from these experiments is that the perceived motion of the target might be playing a role in the saccadic accuracy [Zivotofsky et al., 1996]. When the target velocity is taken into account saccades are correlated with retinal error measured at the moment of target step [Smeets and Bekkering, 2000]. The studies on humans show a great variability in the accuracy of saccades during SPEM [Gellman and Fletcher, 1992; Baker et al., 2003]. However, the source of poor localization is not well understood because there was no correlation found between the pursuit velocity (15°/s, 30°/s, and 45°/s) and the amount of saccadic inaccuracy [Ohtsuka, 1994].

Saccadic suppression

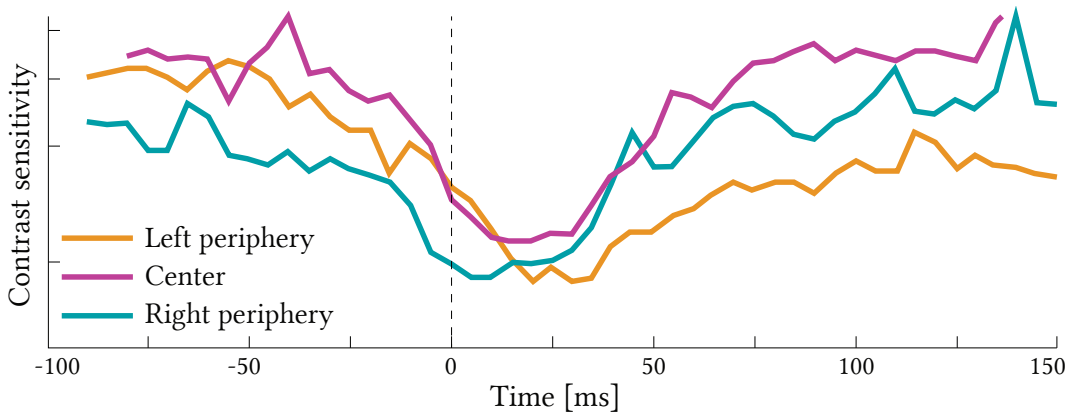


Figure 2.6. Development of the saccadic suppression for three ranges of stimulus positions: left periphery, center, and right periphery. The time is relative to the saccade onset. Plot is adapted from Knöll et al. [2011].

The image of the real world rapidly shifts across the retina during a saccade.

Yet, we do not observe motion blur in the image we perceive due to reduced visual sensitivity [Ditchburn, 1955]. The duration of the reduced sensitivity spans a time interval that starts as soon as 40ms before the start of the saccade and lasts up to 80ms after it ends [Volkmann, 1962; Latour, 1962; Bouman, 1965; Zuber and Stark, 1966; Knöll et al., 2011]. The suppression is characterized by a selective suppression of lower spatial frequencies and the suppression effect decreases as the spatial frequency of the stimulus increases [Volkmann et al., 1978; Burr et al., 1994]. In addition to the reduction in spatial contrast sensitivity, the target position information is also suppressed [Beeler Jr, 1967]. However, the saccadic suppression does not result in perceiving a visual "black-out" due to the visual persistence of retinal images before the saccade [Ritter, 1976; Campbell and Wurtz, 1978]. Figure 2.6 is adapted from Knöll et al. [2011] and shows measurements of how the luminance contrast sensitivity performs during saccade.

Despite the reduced visual sensitivity during the saccades, intra-saccadic perception is still possible. When the saccade peak velocity approximately matches the velocity of sinusoidal gratings rapidly drifting in the same direction, it results in perceived static image of the stimulus for a very brief amount of time during the saccade [Deubel et al., 1987]. Stimulus motion, which is otherwise imperceptible during fixations, can be also perceived during saccades especially when the combined movements of the stimulus and eyes result in retinal frequencies between 10–25Hz [Castet and Masson, 2000]. Based on intra-saccadic perception, an important question is whether saccadic suppression is just a consequence of motion blur in the retinal image or not. Recent studies show that saccadic suppression is not just as a consequence of changes in retinal image and neural activity is also actively suppressed during saccades independent of visual input [Bremmer et al., 2009; Binda and Morrone, 2018].

2.3 Contrast perception

Image contrast is one of the most important features for visual perception [Peli, 1990]. Contrast detection depends on the spatial frequency of a contrast pattern, and it is characterized by *the contrast sensitivity function (CSF)*, illustrated in Figure 2.7, left [Barten, 1999]. The perceived contrast is a non-linear function of contrast magnitude, and the incremental amount of detectable contrast change increases with the contrast magnitude. This effect is often called *self-contrast masking*, and it is modeled using compressive contrast transducer functions [Lubin, 1995; Zeng et al., 2001]. The contrast detection threshold also increases with the neighboring contrast of similar spatial frequency [Legge and

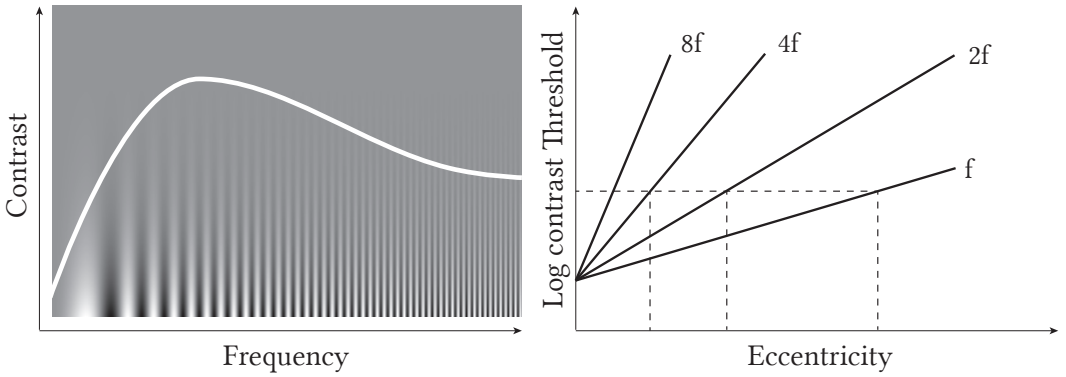


Figure 2.7. Left: An illustration of the contrast sensitivity function. In the area below the white line the observer is able to distinguish the sinusoidal pattern, whereas the area above the line appears completely gray. Note, that the actual position of the function depends on the retinal size of the Campbell-Robson chart and on the luminance of the media used to present it. Right: Hypothetical contrast threshold as a function of eccentricity for various spatial frequencies. Plot adapted from Peli et al. [1991].

Foley, 1980]. To analyze this *spatial masking* effect, often band-pass filter banks are used first to decompose an image into different frequency *channels* [Lubin, 1995; Mantiuk, Kim, Rempel and Heidrich, 2011; Zeng et al., 2001], and then quantify the amount of masking within each channel separately. Majority of perceptual models that are used in various applications, such as image quality evaluation [Lubin, 1995; Mantiuk, Kim, Rempel and Heidrich, 2011], compression [Zeng et al., 2001], and rendering [Bolin and Meyer, 1998; Ramasubramanian et al., 1999], consider all the HVS characteristics mentioned above.

2.3.1 Peripheral vision

Perceptual characteristics of the HVS and, in particular, contrast perception, are not homogeneous over the visual field. This non-homogeneity is related to the non-uniform distribution of retinal sensory cells (Section 2.1.1). To explain the perceptual difference between foveal and peripheral vision, Curcio and Allen [1990] provide anatomical measurements of ganglion cell densities as a function of retinal eccentricity. In more recent work, Watson [2014] parameterizes this relation with a formula for four different visual quadrants and compares the estimations of cell densities with actual measurements from previous studies. Such a parameterization allows computation of the Nyquist frequency for an

arbitrary position in the visual field based on the sampling rate of retinal ganglion cells. However, such anatomical models do not fully explain peripheral sensitivity to visual features, such as contrast. Peli et al. [1991] address this gap and extend the foveal CSF to the peripheral visual field. Although their extension fits well to the previous peripheral contrast sensitivity measurements, it is not a complete model for the peripheral vision. Figure 2.7 illustrates their approach.

2.3.2 Blur sensitivity

The decreased sensitivity to image distortions in peripheral vision motivates foveated rendering techniques (Chapter 3) to save computation time by rendering low-resolution content at larger eccentricities. From the perception point of view, the closest effect extensively studied in the literature is blur perception. For foveal vision, many studies measure detection and discrimination threshold for simple stimuli such as a luminance edge blurred with different Gaussian filters [Watson and Ahumada, 2011]. Similar experiments can be used to measure the sensitivity to blur at various eccentricities [Ronchi and Molesini, 1975; Wang and Ciuffreda, 2005; Kim et al., 2017]. The existing studies reveal a monotonic increase in the threshold values as a function of eccentricity. Unfortunately, simple stimuli used in the above experiments cannot represent the rich statistical variety of complex images. In particular, such threshold values strongly depend on the image content [Sebastian et al., 2015].

Chapter 3

Related Work

This chapter focuses on previous work related to enhancing the gaze-contingent rendering. In particular, Section 3.1 discusses methods for saccade landing position prediction as a tool for combating system latency and improving user experience. Section 3.2 introduces works on foveated rendering and on image metrics looking for intersections where both, the human visual characteristics as well as the underlying image content are considered for improving the visual quality and the computational time.

3.1 Saccade Landing Position Prediction

Saccade velocity and duration cannot be voluntarily controlled, and normally, the oculomotor system follows a preprogrammed ballistic motion trajectory. Although it has been demonstrated that, in some cases, the central nervous system is capable of changing the trajectory of the saccades in flight, it takes approximately 70 ms for visual information to travel from the retina to oculomotor mechanisms of the brain (Chapter 2). Since the duration of saccades usually falls between 20 and 80 ms, there is not enough time to respond to stimuli during the saccade. A large body of work has been dedicated to analyzing saccade ballistics. For example, it has been demonstrated that there is a linear relationship between the duration and the amplitude of saccades [Bahill, Clark and Stark, 1975b]. On the other hand, the same work showed that the relation between the duration and the peak velocity (*the main sequence*) is nonlinear. Moreover, velocity profiles of short saccades are symmetric. This, however, does not hold for medium and long saccades whose profiles are skewed towards their beginning [Van Opstal and Van Gisbergen, 1987].

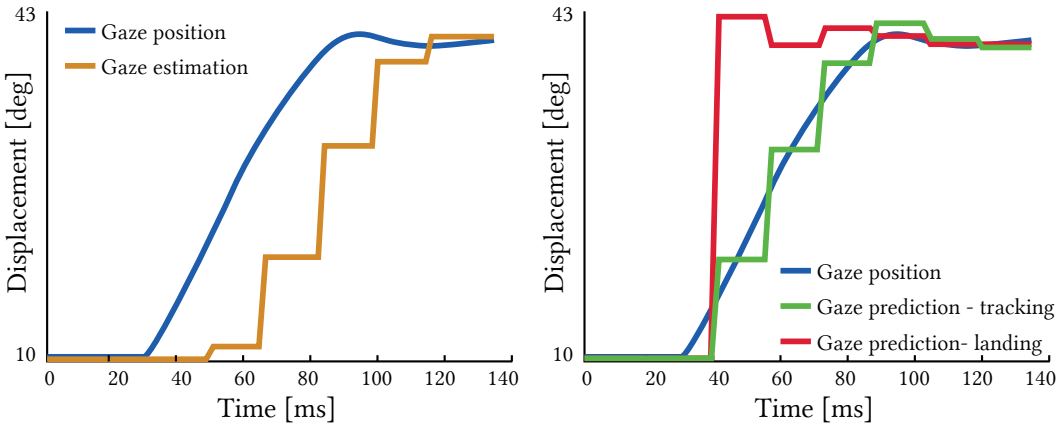


Figure 3.1. Left: Saccade displacement profile (blue) and the gaze estimations used by the system. Notice the latency. Right: Two possible ways to utilize saccade position prediction: either by making prediction for the next frame (tracking) or make a direct prediction for the landing position. Methods for both approaches are introduced by Han et al. [2013a] from where the two plots were adapted.

The ballistic characteristics have been exploited for modeling saccades. Anliker [1976] and Paeye et al. [2016] assume symmetrical velocity profiles, and predict the landing position essentially by doubling the distance traveled until peak velocity. Yeo et al. [2012] define saccade velocity profiles using a bell-shaped curve for simulating HVS dynamics during object tracking. To avoid inaccuracies in velocity estimation a more stable prediction can be provided by fitting a Gaussian function to the gaze velocity [Paeye et al., 2016]. As all of these solutions assume symmetry of the velocity profiles, they can be used only for short saccades [Van Opstal and Van Gisbergen, 1987]. Han et al. [2013a] propose a method based on fitting a compressed exponential function to the eye trajectory. In contrast to our work, they focus on providing short-term predictions (10 ms), while we aim to predict landing positions. Komogortsev and Khan [2009] propose the Oculomotor Plant Kalman Filter (OPKF) that can handle both tasks, and accounts for many anatomical eye properties. A number of anatomy-inspired complex plant saccade models exist, such as [Zhou et al., 2009], but they are less suitable for real-time, gaze-contingent applications as they are not designed to perform prediction [Han et al., 2013a]. A more sophisticated method was proposed by Komogortsev and Khan [2007]. Their Two State Kalman Filter (TSKF) models an eye as a system with position, velocity, and white noise acceleration. In the follow-up work [Komogortsev and Khan, 2009], they extend

TSKF and propose the Oculomotor Plant Kalman Filter (OPKF) which accounts for many anatomical eye properties. Such anatomy-inspired saccade models require many (over 20) adjustable parameters, which make them less suitable for gaze-contingent rendering. Based on the OPKF model, Komogortsev and Khan proposed a computationally efficient chi-square test whose peak value is correlated with the saccade amplitude [Komogortsev, Ryu, Marcos and Koh, 2009]. This model can predict the saccade landing position at an early stage and can be used for fast target selection in gaze-guided computer interaction [Komogortsev, Ryu, Koh and Gowda, 2009]. However, the model was tested only for a single, large (5°) target, and only horizontal saccades were considered. In Komogortsev et al. [2009], an average prediction error of 5.41° was reported for a similar model, where again only horizontal saccades were considered, while the model proposed by Anliker [1976] performed best in such conditions with an average error of 3.46°.

Following the results achieved in Chapter 4, in order to combat system latencies typically observed in gaze-contingent rendering systems, Griffith et al. [2019; 2020] proposed the use of support vector machine regression models for saccade landing position prediction and showed an extension to oblique saccades. Later, Morales et al. [2018; 2021] proposed the use of Long Short-Term Memory (LSTM) networks for saccadic landing position prediction and Griffith et al. [2020] introduced a technique to improve the performance of LSTM and feed-forward network based models. Despite these active research efforts in saccade prediction for gaze-contingent rendering, investigation of different factors and their influence on the saccade landing position prediction, as well as incorporating them into a real-time rendering system remains an open problem.

3.2 Foveated Rendering

The fact that the sensitivity of the HVS to luminance contrast, color, and depth is concentrated in the fovea and declines significantly towards the periphery (Chapter 2) can be beneficial in rendering systems that use eye tracking devices. Such systems utilize gaze-contingent techniques to change a set of functions in their pipeline depending on the gaze location. Foveated rendering is such a technique, where the key idea is to increase rendering performance by providing lower quality to the peripheral vision and keeping the quality unchanged for the foveal vision. Traditional techniques (Section 3.2.1) exploit only these characteristics of the HVS while image metrics (Section 3.2.2) lay foundation for content-dependent techniques (Section 3.2.3), which consider the underlying

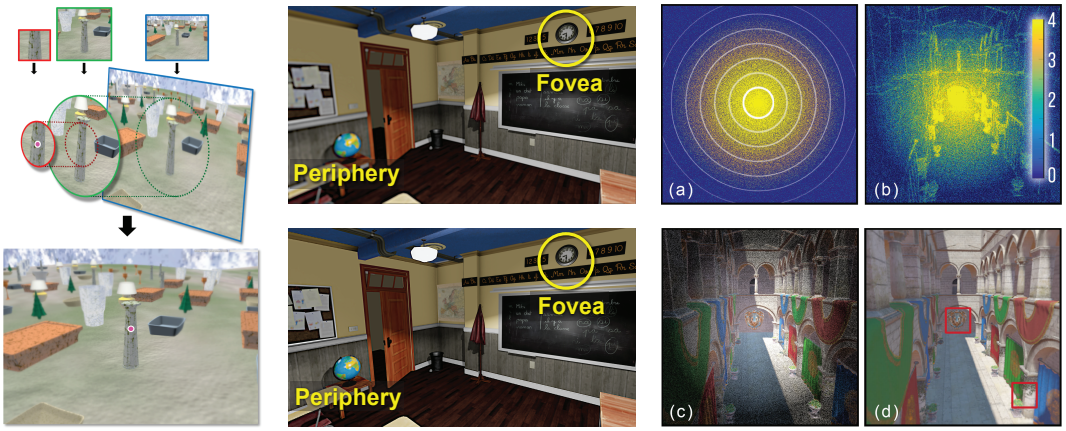


Figure 3.2. Left: Standard foveated rendering using three layers of a single frame, rendered in various resolutions and sizes. Image from Guenter et al. [2012]. Middle: Perceptually-enhanced foveated rendering. Image from Patney et al. [2016]. Right: Perceptually-adapted sampling pattern for foveated sparse shading. Image from Stengel et al. [2016].

image content and the way the HVS responds to it.

3.2.1 Traditional techniques

Gaze-contingent rendering has many potential applications focused on the improvement of viewing experience and reduction of the computation costs. Gaze-driven solutions contribute to the improvement of tone mapping [Jacobs et al., 2015], depth perception [Kellnhofer et al., 2016] and viewing comfort in stereoscopic displays [Duchowski et al., 2014]. Computational depth-of-field effects partially compensate for the lack of proper eye accommodation in standard displays [Mantiuk, Bazyluk and Tomaszewska, 2011; Mauderer et al., 2014], while for displays with accommodative cues, proper alignment of multi-focal images can be achieved [Mercier et al., 2017] or laser beams can be guided by pupil tracking [Jang et al., 2017]. The computation performance may be improved by reducing the level of detail [Reddy, 2001; Duchowski et al., 2009], or spatial image resolution [Guenter et al., 2012; Vaidyanathan et al., 2014; Swafford et al., 2016] towards the periphery, which is particularly relevant for this thesis. In their work Guenter et al. [2012] implement the peripheral fall-off by rendering a single frame with three different resolution, with higher resolutions enclosing only a part of the frame. These frames are then merged in order to present the higher resolution only around the gaze location whereas the lower resolutions

fill in the periphery (Figure 3.2, left).

3.2.2 Image metrics

Perceptual experiments studying the sensitivity of the HVS to contrast changes can be used for developing image metrics which are then used to evaluate or drive image synthesis techniques. In this context, Watson and Ahumada [2011] argue that when the reference and blurred images are given as inputs, general models of contrast discrimination can account for blur perception for simple stimuli in the fovea. Their model works by summing the energy over a restricted local extent and uses the CSF as well as the spatial contrast masking effects. Sebastian et al. [2015] employ a similar generic model to predict their data for complex images, while Bradley et al. [2014] additionally consider local luminance adaptation to account for near eccentricity (up to 10°). The work by Swafford et al. [2016] extends the advanced visible difference predictor HDR-VDP2 [Mantiuk, Kim, Rempel and Heidrich, 2011] to handle arbitrary eccentricities by employing a cortex magnification factor to suppress the original CSF. The authors attempt to train their metric based on data obtained for three applications of foveated rendering, but they cannot find a single set of parameters that would fit the metric prediction to the data. In a work, which follows the results of this thesis, [Mantiuk et al., 2021] design FovVideoVDP - a video difference metric that models simultaneously the spatial, temporal, and peripheral aspects of the human perception.

3.2.3 Content-dependent techniques

Image content analysis to improve quality and efficiency in foveated rendering has been considered to a relatively limited extent. Patney et al. [2016] use contrast enhancement to help recover peripheral details that are resolvable by the eye but degraded by filtering that is used for image reconstruction from sparse samples (Figure 3.2, middle). Stengel et al. [2016] make use of information available from the geometry pass, such as depth, normal, and texture properties, to derive local information on silhouettes, object saliency, and specular highlights (Figure 3.2, right). The combined features along with visual acuity fall-off with the eccentricity and luminance adaptation state (based on the previous frame) allow for sparse sampling of costly shading. As luminance information is not available before shading for the current frame, contrast sensitivity and masking cannot be easily considered. Sun et al. [2017] propose a foveated 4D light field rendering with importance sampling that accounts for focus differences between

scene objects, which are determined by the object depth and the eye accommodation status at the fixation point. This leads to the reduction of computation costs for optically blurred scene regions, which requires displays that can trigger the eye accommodation.

Chapter 4

Saccade Landing Position Prediction for Gaze-Contingent Rendering

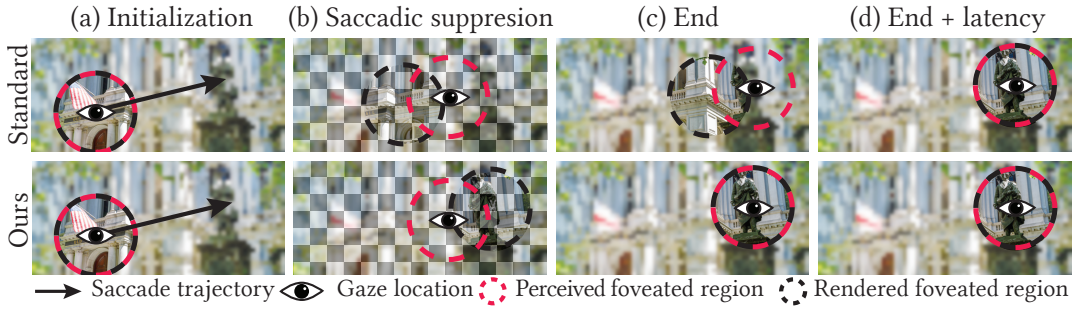


Figure 4.1. Standard gaze-contingent rendering (top row) updates the image according to the current gaze prediction. Due to the system latency, during a saccade, there is a significant mismatch between the rendering and the actual gaze position (b, c). The method moves the foveated region to the actual gaze position only after a delay equal to the system latency (d). Our technique (bottom row) predicts the ending position of the saccade at its early stage and updates the image according to the new prediction as soon as it is available (b). Due to the saccadic suppression the user cannot observe the image manipulations during the saccade (b). When the saccade ends and the suppression is deactivated (c), the observer sees the correct image at the new gaze position with our method.

Despite the constant improvement of hardware, meeting the quality demands regarding spatial and temporal resolutions, stereoscopic presentation, and scene complexity required in current applications is still a challenging problem. This is

manifested in the recent developments of new mobile platforms as well as virtual and augmented reality (VR/AR) systems, where both the quality and energy efficiency are limiting factors that have to be tackled to enable full adoption of these technologies. With the recent advances in affordable eye-utracking technology, the above problems can be addressed by exploiting properties of the human visual system (Section 2.1). The most prominent example is foveated rendering techniques (Section 3.2) that provide high image quality only for the fovea. This leads to improvements both in the rendering time and quality.

Although gaze-contingent rendering can lead to significant improvements, it is very sensitive to system latency [Saunders and Woods, 2014]. Even short delays may result in visible artifacts which make the gaze-contingent rendering unfavorable (Chapter 1). Therefore, displaying an updated image after saccade completion is critical, and any delays may limit the benefits of gaze-contingent rendering. In practice, to prevent problems with system latency, high-end equipment has to be used (Section 3.2). While such equipment is becoming a commodity, high-quality rendering rarely achieves sufficient frame rates. As a result, the gaze-contingent systems introduce significant latency which leads to visible artifacts such as the perception of low-quality image from the peripheral rendering.

To address this problem, we propose a new technique for controlling gaze-contingent rendering during saccades. The key idea is to maximally exploit the saccadic suppression when it is the strongest, i.e., during saccades. Instead of placing the foveated region at each gaze position as soon as it is provided by the eye tracker, our method fixes it to a predicted saccade landing position (Figure 4.1). The prediction is continuously adjusted during the saccade for new gaze direction samples so that the delay with which the correct image appears is minimized when the saccade ends. During the saccade, the mismatch between the actual gaze direction and the rendering is hidden by the saccadic suppression. In this dissertation, we propose a method for predicting the saccade landing position. It accounts for both within- and between-participant saccade variability as well as inaccuracies of modern eye trackers. We demonstrate how the prediction can be used in the context of gaze-contingent rendering to alleviate the problem of system latency. Our user experiments validate the accuracy of the predictions and the quality improvements when our strategy is applied. To provide further insights, we use different combinations of display frame rate and eye-tracking sampling rate in our tests. We present the following contributions:

- measurements and analysis of saccade trajectories,
- a new model for predicting the landing position of saccades,

- a comparison of several prediction techniques based on our measurements, and
- two experiments which validate both a subjective and an objective quality increase when our method is applied in a gaze-contingent rendering system.

4.1 Overview

In our modeling, we aim to predict the landing position of saccades to update gaze-contingent rendering early enough to reduce the delays coming from the latency of the system. The greatest challenge in building such a system is robustness. Standard gaze-contingent rendering suffers from latency; however, the introduced delay is always consistent. When a prediction is used, even small errors may lead to catastrophic failures which will result in clearly visible artifacts, and therefore, user dissatisfaction. Our goal is to create a system which is robust to fluctuations of eye-tracking data arising from instabilities in eye movements, as well as from eye-tracker errors.

To this end, we follow the assumption that saccades obey ballistic trajectories which are determined mainly by saccade amplitude. Even though this assumption does not hold completely, as other factors may influence saccades (Section 4.3), we demonstrate that it leads to a simple yet powerful and robust model that provides significant improvements in gaze-contingent applications (Section 6.6). To gain knowledge about saccade characteristics, we first perform measurements to collect samples of many saccades performed by several participants (Section 4.1.1). After analyzing the collected data (Section 4.1.2), we construct a computational model (Section 6.3) that captures the characteristics of different saccades and uses eye-tracker samples to predict their landing positions.

4.1.1 Measurements

We conducted a user experiment to collect a large amount of data associated with different saccades. To evoke saccades, participants were asked to focus on a target stimulus, which was a white dot on a uniform 50%-gray background shown on a screen operating at 60 FPS. The target changed its position when the user pressed a key after fixation. The target positions were pre-generated and shuffled so that different saccade amplitudes, spanning a range of 5°–45°, were

equally represented in the collected data. The data collection was performed with an eye tracker at 300 Hz sampling frequency. The eye tracker allows free head movement during tracking; nevertheless, we used a chin-rest to improve the tracking accuracy. The viewing distance was fixed to 70 cm, which resulted in a coverage of 46×27 visual degrees. 22 participants with normal or corrected-to-normal vision took part in our measurements. The eye tracker was calibrated for each participant. Every participant performed at least 300 saccades which were recorded during a 5-minute session. Figure 4.2 shows our setup and gaze data recorded from one of our participants.

System Details In all our experiments, including the validation described in Section 6.6, we used a Tobii TX300 eye tracker, capable of 300 Hz, 120 Hz and 60 Hz sampling rates. The display was a 27" 2560 × 1440 ASUS ROG Swift PG278Q which supports 60 Hz, 120 Hz and 144 Hz refresh rates. The CPU and GPU of our platform were an Intel(R) Xeon(R) E5-1620 v3 @ 3.50 GHz and NVIDIA GeForce GTX 660, respectively. The software system used was our own C++ implementation based on OpenGL.

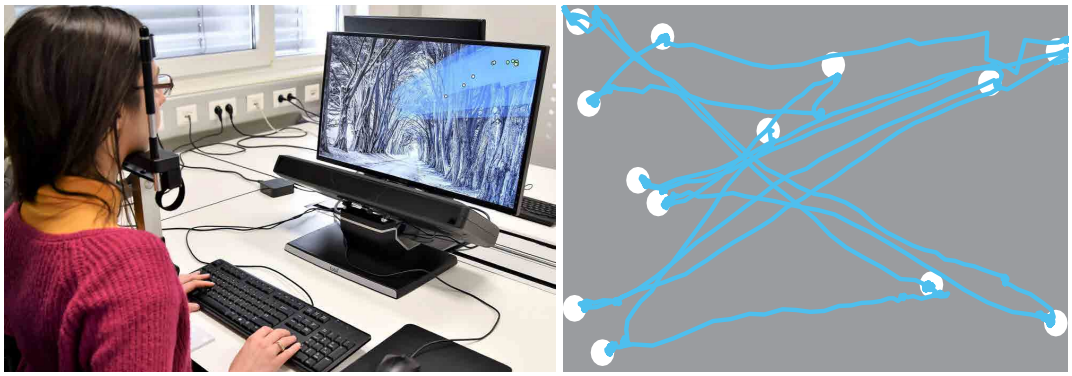


Figure 4.2. Our measurement setup and a sample subset of eye-tracking data from one participant. The white circles visualize saccade targets shown during the experiment, whereas the blue traces correspond to eye-tracker samples. For simplicity, only 12 targets are visualized here. Throughout the actual experiment, 300 consecutive targets were shown to each observer.

4.1.2 Data Processing

The data recorded in the experiments includes both saccades and fixations. To build our model, we need to extract eye-tracking samples corresponding to saccades. Detecting saccades can be easily performed in a post-processing step by

analyzing velocity profiles. However, in order to be consistent with how the saccades will be detected in gaze-contingent rendering techniques, we opt for a robust analysis that is suitable for on-line data. Several techniques have been proposed and analyzed in this context [Salvucci and Goldberg, 2000; Andersson et al., 2016]. For our purpose, we found that the velocity threshold method (I-VT) provides satisfactory results. I-VT relies on the fact that the saccades are very fast eye movements and detects them as soon as the high velocity of eye movement is observed. We denote the detection threshold as V_d and refer to the first gaze sample whose velocity exceeds V_d as *the detection point*.

For the saccade detection to be robust, V_d has to be relatively high, usually above $100°/s$. This means that the true beginning of the saccade is much earlier than the detection point. To obtain all the past samples of the saccade, we scan the gaze samples backward in time to find the beginning. Due to the inherent noise of eye trackers and small movements of the eye, the velocity is always positive even during fixations. Therefore, we employ a two-step procedure similar to Dorr et al. [2010] by introducing another velocity threshold, V_a. The first sample where the measured gaze velocity is equal to V_a is the *anchor point*, i.e., beginning of the saccade. Due to discrete sampling of eye trackers, it is unlikely that a sample with this exact velocity will be found in practice. Therefore, we introduce an additional sample for velocity V_a, by interpolating between the first sample for which the velocity is over V_a and the previous sample. We also use a velocity threshold to detect the end of the saccade and refer to the first sample whose velocity drops below V_f as *the end point*. Additionally, we treat samples occurring up to 15 ms after *the end point* as a part of the saccade to account for potential corrective saccades called *glissades*, which are typically not larger than 0.5°–2° [Holmqvist et al., 2011, Ch. 2]. We ignore the *glissades* detected separately by removing the detections which are shorter than 15 ms. For increased robustness, we only take the saccades whose *anchor points* are found within a 30 ms interval before the detection. If tracking is lost during a saccade, it is not used for training our model. Please see Figure 4.3, which shows the anchor, detection, and end points for a sample saccade.

We define each saccade, S_k, as a set of subsequent gaze samples from the eye tracker:

$$S_k = \{s_{k0}, s_{k1}, s_{k2}, ..., s_{kN}\}, \tag{4.1}$$

where s_{k0} is the gaze sample corresponding to *the anchor point* of the saccade and s_{kN} is *the end point* of the saccade. The gaze samples are expressed in terms of triplets:

$$s_{kl} = \langle t_{kl}, d_{kl}, \theta_{kl} \rangle, \tag{4.2}$$

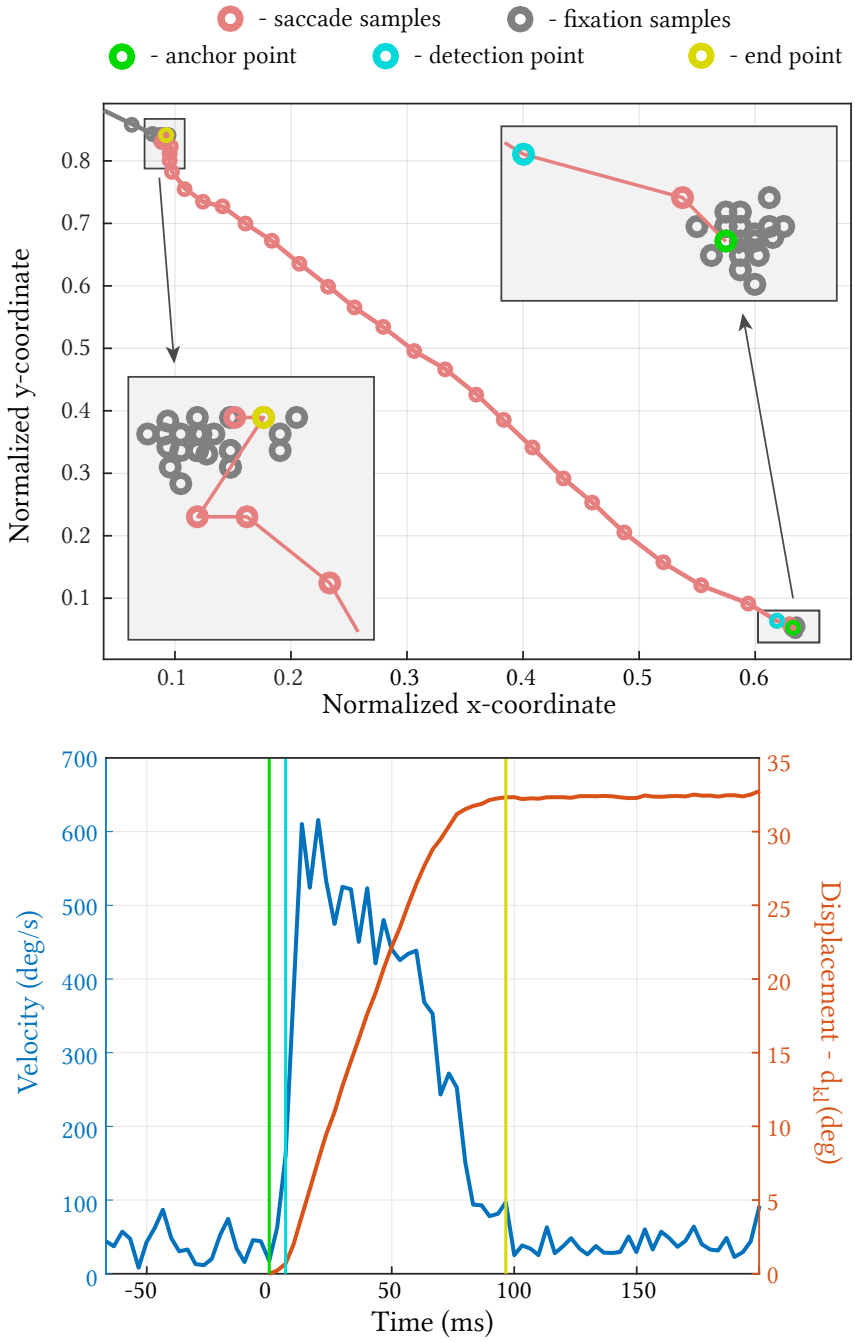


Figure 4.3. Visualization of our data processing for one sample saccade. Top: Gaze samples for a saccade from the bottom-right corner to the top-left corner of the screen. Bottom: The gaze velocity and displacements. Samples corresponding to the anchor, detection and the end points are indicated in green, cyan and yellow colors, respectively. Captured with a Tobii TX300 eye tracker at 300 Hz sampling frequency.

where t_{kl} is the timestamp, d_{kl} is the displacement and θ_{kl} is the direction of the sample. t_{kl}, d_{kl} and θ_{kl} are measured with respect to *the anchor point*; therefore,

$$t_{k0} = 0,\ d_{k0} = 0,\ \theta_{k0} = 0. \tag{4.3}$$

This representation is similar to the polar coordinate system where the origin is *the anchor point* of the saccade and d_{kl} corresponds to the radial coordinate, while θ_{kl} corresponds to the polar angle. d_{kl} is measured in terms of visual degrees instead of pixels to make it independent of the distance between the observer and the screen.

Given our representation, the amplitude of saccade S_k is defined as

$$|S_k| = d_{kN}, \tag{4.4}$$

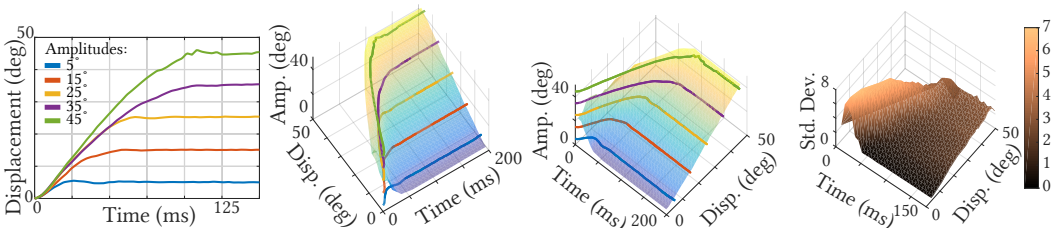
and the direction as

$$\angle S_k = \theta_{kN}. \tag{4.5}$$

The choice of velocity threshold is crucial for robustness to noise and small involuntary eye movements. Saccades involve very fast eye motion; therefore, they achieve speeds that cannot be observed during other eye movements. Usually, the achieved velocity during a saccade exceeds $100°/s$. Consequently, we set $V_d = 130°/s$. Possible misses of very short saccades are not very problematic as these usually do not lead to problems in gaze-contingent rendering. Also smooth-pursuit eye movements are successfully filtered out by this detection threshold as they rarely reach velocities above $80°/s$ [Meyer et al., 1985; Daly, 1998]. For the anchor point, the velocity threshold V_a needs to be chosen such that we do not include eye-tracking samples corresponding to fixations. We observed that a good choice of the threshold may depend on the participant, as different eye-tracking noise and involuntary movements can be observed for different participants. However, for better generalization, we decided to choose a conservative value that reduces the probability of including fixation samples as saccades in our training data. Consequntly, we set $V_a = 60°/s$, which is used in all our experiments. We set the threshold for the end point accordingly as $V_f = 60°/s$. All our choices remain in agreement with the general characteristics of saccades' velocity profiles described by Boghen et al. [1974b].

4.1.3 Model

The problem of predicting the landing position can be defined as estimating $|S_k|$ and $\angle S_k$ for every timestamp t_{kl} during the saccade, i.e., before d_{kN} and θ_{kN}



(a) (b) (c) (d)

Figure 4.4. Displacement profiles of one participant for different saccade amplitudes are given in (a). The corresponding prediction surface is shown from two different viewing angles in (b) and (c) with the saccade amplitude in the z-axis. Standard deviation of the prediction surface across all participants is given in (d). For simplicity, only 5 individual saccades are shown in (a), (b) and (c). We collected more than 300 saccades from each participant.

are actually observed. As the trajectories of most saccades are linear or approximately linear [Leigh and Zee, 2015b, Ch. 3], our estimate for the direction of the saccade, $\widehat{\angle S_k}$, at a point with timestamp t_{kl} is equal to the direction of the last observed gaze sample:

$$\widehat{\angle S_k}(t_{kl}) = \theta_{kl}. \tag{4.6}$$

Because of the different acceleration, deceleration, peak velocity and duration characteristics, the displacement observed at a specific time after *the anchor point* depends on the amplitude of a saccade. Figure 4.4a shows how displacement profiles change with respect to different saccade amplitudes. The characteristics of this change are captured in the data that we collected in our experiment (Section 4.1.1). The displacement profiles tend to form a consistent surface in the 3D space where the x- and y-axis are the displacement and time axes, while the z-axis corresponds to the amplitude values (see Figure 4.4b-4.4c).

We treat the amplitude prediction as a regression problem. Based on the collected data, we seek a function f that estimates the saccade amplitude given a displacement d_{kl} at a given time period t_{kl} from the beginning of the saccade to the current samples. Formally, we define our prediction as:

$$\widehat{|S_k|}(t_{kl}) = f(t_{kl}, d_{kl}), \tag{4.7}$$

and require function f to minimize:

$$\sum_{k,l} (|S_k| - f(t_{kl}, d_{kl}))^2. \tag{4.8}$$

While the amplitude function $f(t_{kl}, d_{kl})$ gives us a direct prediction of saccade length, it is also possible to model the displacement during saccades as a

function of time and amplitude. This may potentially provide a more stable solution, as modeling the steep part of f for small values of displacement and time (Figure 4.4) might be problematic. To this end, we also consider a function g that minimizes:

$$\sum_{k,l}(d_{kl} - g(|S_k|, t_{kl}))^2. \tag{4.9}$$

The prediction of saccade length using g requires a linear search that, for a given timestamp t_{kl} and displacement d_{kl}, finds $\widehat{|S_k|}$ such that $\widehat{|S_k|} = g(|S_k|, t_{kl})$. More formally,

$$\widehat{|S_k|}(t_{kl}) = \arg\min_x (d_{kl} - g(x, t_{kl}))^2. \tag{4.10}$$

Both Equations 4.7 and 4.10 perform prediction based only on the last sample from the eye tracker. This does not account for information provided by all saccade samples. To investigate whether all samples can provide a better prediction, we modify the prediction in Equation 4.10 to account for all the samples:

$$\widehat{|S_k|}(t_{kl}) = \arg\min_x \sum_{n=1}^{l} (d_{kn} - g(x, t_{kn}))^2. \tag{4.11}$$

Effectively, this prediction takes all saccade samples observed until time t_{kl} and tries to find $\widehat{|S_k|}$ such that the saccade profile best fits function g.

Both of the functions f and g can be realized using different parameterization techniques. In order to find the best technique, we compare predictions provided by a polynomial fitting and an interpolation. To this end, for each method, we treat every participant separately and find a personalized polynomial fit or an interpolation grid that minimizes Equation 4.8 or Equation 4.11. To determine the degree of the polynomials and to avoid over-fitting, we analyze cross-validation errors for polynomial degrees ranging from 1 to 7. According to this analysis, polynomial degrees 5 for t and 2 for d provide the lowest cross-validation error for the largest number of participants. Although it is possible to optimize the polynomial degrees for each participant separately, this procedure did not provide better fits in our experiments. Consequently, we used a fixed degree for all polynomial-based models. To enable interpolation, we remove multiple measurements corresponding to the same pairs (t_{kl}, d_{kl}) by applying a 2D median filter on saccade amplitudes. This step also reduces the effect of noise and outliers. The size of the filter is 0.5° for the displacement and 2 ms for the time dimension.

We use 80% of our saccade measurement data (Section 4.1.1) to build prediction models for all methods described above. Later, we use the remaining

20% for testing to compare their performance. The mean absolute error as well as the standard deviation of the error are shown in Figure 4.5 for all prediction methods. Both the prediction accuracy and standard deviation of the prediction error improve during the course of the saccade. This, as expected, is a result of diverging behavior of displacement profiles for different saccade amplitudes (Figure 4.4a). The best performing method in terms of error and standard deviation is the interpolation based on Equation 4.7 whose mean error drops below 4° in the middle of the saccade and is less than 1° at 80% of the normalized saccade time. This magnitude of prediction error is significantly smaller than the typical foveal region size in gaze-contingent rendering [Patney et al., 2016, Fig. 8]. Interestingly, the best prediction is based on only one (the last) sample from the eye-tracking data, which means that our prediction does not benefit from including all of the saccade samples. This might be related to less reliable information provided by the samples at the early stage of the saccades. In addition, we measure the variability of interpolation-based model between participants (see Figure 4.4d). The high variance at the beginning of the saccades suggests that their characteristics differ significantly between subjects, which supports the choice of using personalized models.

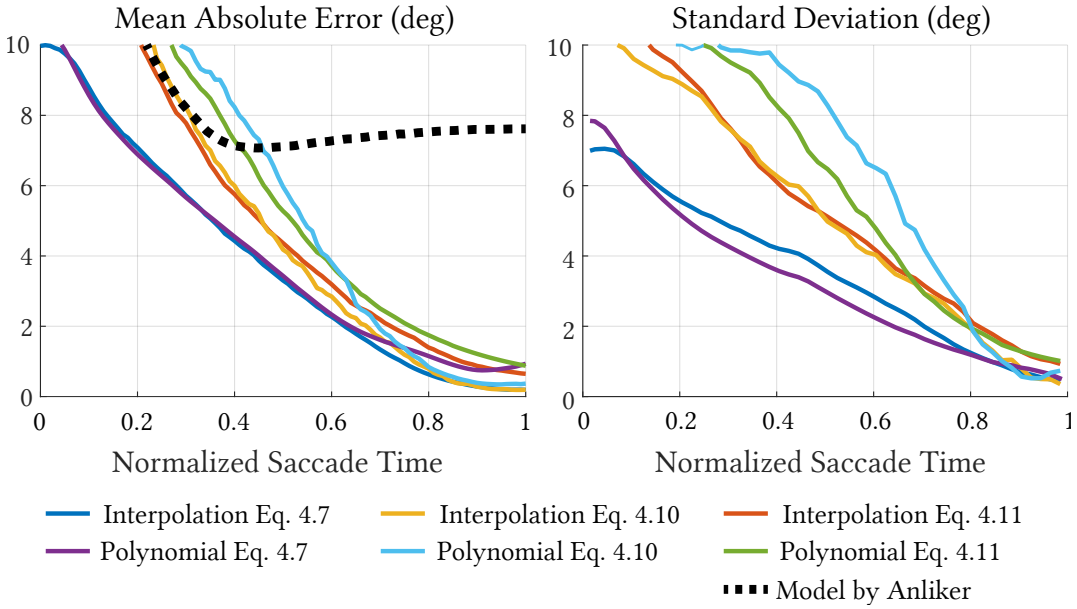


Figure 4.5. Comparison of different prediction modeling methods. Left: the mean absolute error as a function of normalized saccade time. Right: the standard deviation of the mean absolute error.

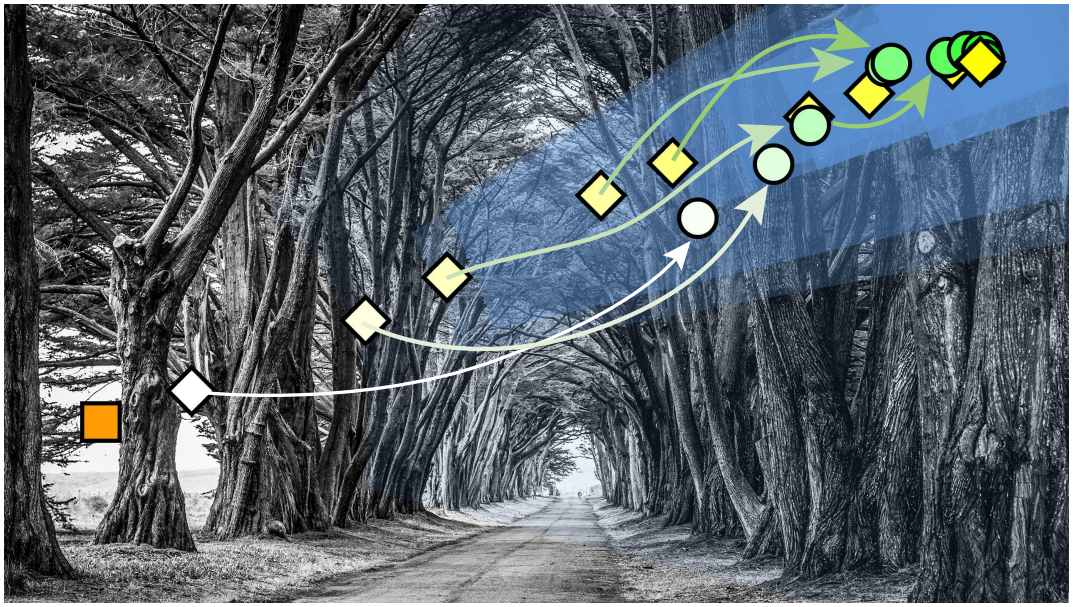


Figure 4.6. Gaze samples (yellow), landing position predictions (green) and corresponding prediction intervals (blue) for a sample saccade. The beginning of the saccade (orange), gaze samples and predictions are shown as the square, diamond and circular shapes, respectively. Color saturation level of the points indicates the time when each sample is observed and each prediction is made (more saturated color indicates more recent sample and prediction). Arrows connect gaze samples with our model's corresponding predictions for the landing position.

We compared the prediction performance of our method with that of Anliker [1976]. We observe that this method results in high prediction errors caused by severe undershooting for large saccades due to the violation of symmetry in velocity profiles (Figure 4.5, left). This suggests that any prediction method based on the symmetry assumption (including symmetric curve fitting approaches [Paeye et al., 2016]) would likely suffer from the same type of inaccuracy.

Average Model In addition to personalized models, we further investigated the possibility of replacing them with one averaged model which gives the flexibility of predicting saccade amplitudes without the training step. To derive the average models, we used the same procedure as for the personalized models, but with a leave-one-out cross-validation strategy which removes one participant for the training stage. Figure 4.7 shows how much personalized models improve the mean absolute error compared to the average model both for interpolation

and the polynomial fit approach. As expected, for most of the participants personalized models provide better prediction, by up to 30% (1°). This might be explained by the high variance between pesonalized models (Figure 4.4d). Interestingly, the personalized model of one participant performs about 10% worse than the average model, which might be related to the high measurement noise observed during the data recording session for this participant. The comparison suggests that although in most cases the personalized model leads to improved predictions, the average model is a practical alternative. Here we report the average model that results from the polynomial fit approach:

$$\begin{aligned} f_{poly}(t, \acute{d}) = & -10.19t + 19.11t^2 - 17.15t^3 + 6.251t^4 - 0.7552t^5 \\ & - 23\acute{d}t + 26.89\acute{d}t^2 - 14.74\acute{d}t^3 + 3.882\acute{d}t^4 - 0.4005\acute{d}t^5 \\ & - 11.84\acute{d}^2t + 9.326\acute{d}^2t^2 - 2.583\acute{d}^2t^3 - 0.01058\acute{d}^2t^4 \\ & + 0.06587\acute{d}^2t^5 + 20.18\acute{d} + 5.071d^2 + 18.15, \end{aligned} \tag{4.12}$$

where $t = (t_{kl} - 47.39)/33.4$ and $\acute{d} = (d_{kl} - 14.67)/11.72$ are the time and displacement measurements normalized with their respective means and standard deviations from the training data.

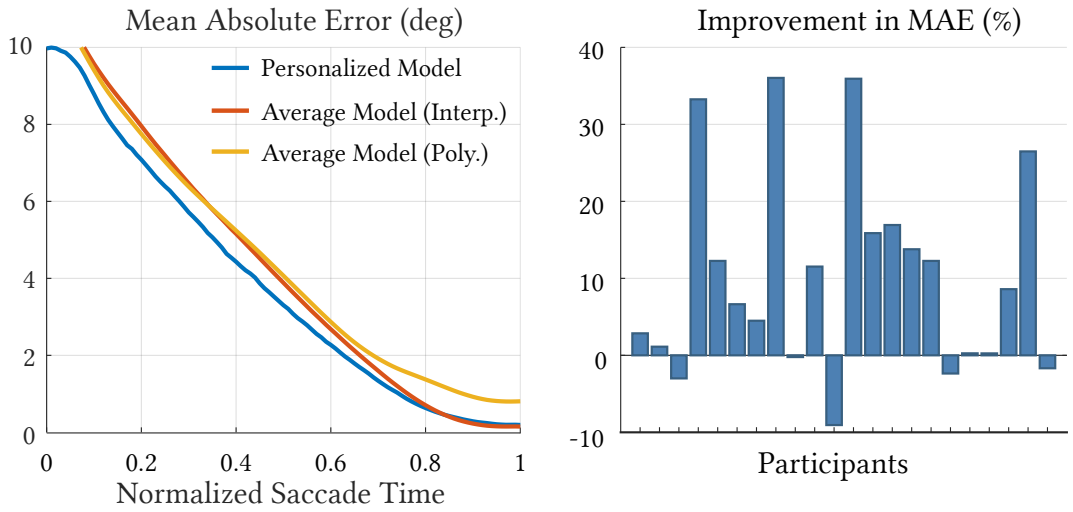


Figure 4.7. Left: The mean absolute error of personalized models is compared with that of the average models using interpolation and polynomial fitting. Right: The amount of improvement in the mean absolute errors when personalized models are used instead of the average model for each participant.

Prediction Intervals Saccades may exhibit variance due to the noise in the firing of motoneurons, target detection inaccuracy of the participant, and measure-

ment inaccuracies [Smeets and Hooge, 2003] (Section 4.3). The information about the within-participant variance is captured in our data. We compute 95th-percentile prediction intervals over the interpolation grid for both the direction and amplitude predictions. Figure 4.6 demonstrates the prediction intervals for each prediction made during the saccade. Please note how the size of the intervals gets small very quickly. Although we use these intervals only for visualization purposes, we believe that they provide additional information that could be used in the gaze-contingent rendering or gaze-driven interaction techniques.

Application of the Model To apply our prediction during a gaze-contingent rendering, we replace eye-tracker samples that correspond to saccade samples with our prediction. To detect saccade samples, we apply the same strategy as described in Section 4.1.2. We start using our technique as soon as the saccade is detected at sample s_{k1}. From that point, all predictions are accepted until we detect the end of the saccade, making s_{kN-1} the last substituted sample. After this sample, we switch to the standard gaze-contingent rendering method which directly uses the samples from the eye tracker. To make the method more robust to noise, we exclude saccades for which the direction changes by more than 5° for the first three consecutive gaze samples at the 300 Hz sample rate and by more than 12.5° at the 120 Hz sample rate. In such cases, the detection is regarded as a false positive, and we switch to the standard gaze-contingent rendering which updates the position of the foveated region according to the most recent gaze sample. For all the samples that are classified as saccade samples, we compute the prediction based on the predicted direction (Equation 4.6) and the amplitude (Equation 4.7 and 4.10). The predicted saccade amplitude $\widehat{|S_k|}(t_{kl})$, and direction $\widehat{\angle S_k}(t_{kl})$, are transformed into a vector representing 2D screen coordinates as follows:

$$p(t_{kl}, d_{kl}) = \Big[s_{k0}^{(x)} + h\Big(\widehat{|S_k|}(t_{kl})\Big)\cos\Big(\widehat{\angle S_k}(t_{kl})\Big), \\ s_{k0}^{(y)} + h\Big(\widehat{|S_k|}(t_{kl})\Big)\sin\Big(\widehat{\angle S_k}(t_{kl})\Big)\Big]^{\intercal}, \tag{4.13}$$

where $s_{k0}^{(x)}$ and $s_{k0}^{(y)}$ are the horizontal and vertical coordinates of the *anchor point* on the screen, respectively, and $h(.)$ is the function which converts the displacement in visual angles to pixel displacement on the screen plane. Next, instead of using the current eye-tracker sample for updating gaze-contingent rendering, we use our prediction. We store the prediction model as an interpolation look-up table or polynomial coefficients for each participant. The whole prediction adds a negligible cost to the gaze-contingent method as it involves a simple lookup and interpolation, or evaluating a polynomial.

4.2 Validation

To validate our strategy for updating gaze-contingent rendering, we performed two user experiments. In the first one, we used simple, synthetic stimuli to demonstrate that our prediction can significantly reduce the delay in updating gaze-contingent rendering after saccades. The second experiment demonstrates a more natural scenario when foveated rendering is used to render natural scenes. By allowing the user to freely explore the content, we measured the influence of our technique on the user experience. Furthermore, we validated our technique for three different combinations of display frame rates and eye-tracker sampling frequencies to investigate the influence of system latency on the performance of our method. The viewing setup as well as the hardware used in these experiments was the same as in our measurements (Section 4.1.1).

4.2.1 Guided-Viewing Experiment

The goal of the first experiment was to demonstrate that our technique can lead to quicker updates of the foveated region. To this end, we designed a simple experiment which was inspired by foveated rendering techniques. The stimulus consisted of four differently oriented Landolt C shapes arranged in a 2×2 grid. Three of them were sharp, and one was blurred (Figure 4.8, top). The size of each shape was 0.4°, the entire 2×2 grid was 1.2°, and the standard deviation of the Gaussian blur was 0.04°. During the experiment, the stimuli appeared in different locations. Each time the orientations of the Landolt C shapes (up, down, left, right), as well as the position of the blurred one, were chosen randomly. Each time the gaze prediction was in some proximity to the stimulus center (7°) the blur was removed to reveal the masked shape. In half of the cases, appearing in random order, the gaze sampling from the eye tracker was used directly; in the rest of the cases, our prediction was used. The task of the participants was to indicate the orientation of the Landolt C shape that was blurred. After receiving the choice of the participant, a new stimulus was displayed at a different position. The idea was that any delays in gaze-contingent rendering would result in an associated delay in removing the blur. Therefore, the longer the delay in the image update, the easier it was to spot which shape was blurred.

This type of blur-removal strategy simulates the foveated region update scheme used in gaze-contingent rendering. If the update lags behind the actual gaze position as in the standard gaze-contingent applications, the participant will arrive at the stimulus position before the blur is removed and will be able to indicate the

shape that was blurred. On the other hand, if our method is used, the foveated region will already be moved to the stimulus position before the arrival of the observer; therefore, the position of the blur will be harder to determine.

Nine participants with normal vision took part in this experiment. For each participant, we used his personalized model to predict saccade landing positions. Each participant saw 300 stimuli for which they indicated the orientation of the blurred Landolt C shape. The experiment took approximately five minutes. We used an eye-tracking sampling frequency and display frame rate of 120 Hz and 60 FPS, respectively. Figure 4.8 (right) shows the ratio of correct responses averaged across all participants. For presentation purposes, we grouped the data according to the amplitude of saccades. For all cases, the number of correct answers was lower when our prediction was used. This indicates that for our update strategy, the blur was removed earlier, effectively reducing the influence of the system latency. The results for medium and large saccades are statistically significant according to Fisher's exact test ($p < 0.001$). For large saccades, the ratio of the correct responses for our method is very close to the expected value of purely random choice (0.25). The difference between the two update strategies is not significant for the shortest saccades. Probably this is related to the proximity of the stimulus to the foveated region, i.e., the stimulus that appears very close to the current fixation location is already in the foveated region and no blur is applied.

The results of this experiment demonstrate that our prediction is fast and accurate enough to reduce the influence of the system latency causing delays in gaze-contingent rendering. As we were not able to demonstrate an advantage for our technique for the shortest saccades, in the rest of our experiments, we modify our prediction model to respond only for medium and long saccades. This is done by raising the detection velocity threshold V_d to $180°/s$. This, in accordance with Boghen et al. [1974b], eliminates saccades shorter than 10°.

4.2.2 Free-Viewing Experiment

The first experiment confirms that our prediction technique allows us to move the region with high rendering quality to the new fixation locations sooner than standard gaze-contingent rendering. However, this does not guarantee that our technique is free of any artifacts and that it provides better perceived quality in general. Therefore, in the second experiment, we validated the performance of our technique in a free-viewing scenario. To this end, we implemented a simple method that imitates a standard foveated rendering technique. Inspired by recent gaze-contingent rendering techniques [Guenter et al., 2012; Patney et al.,

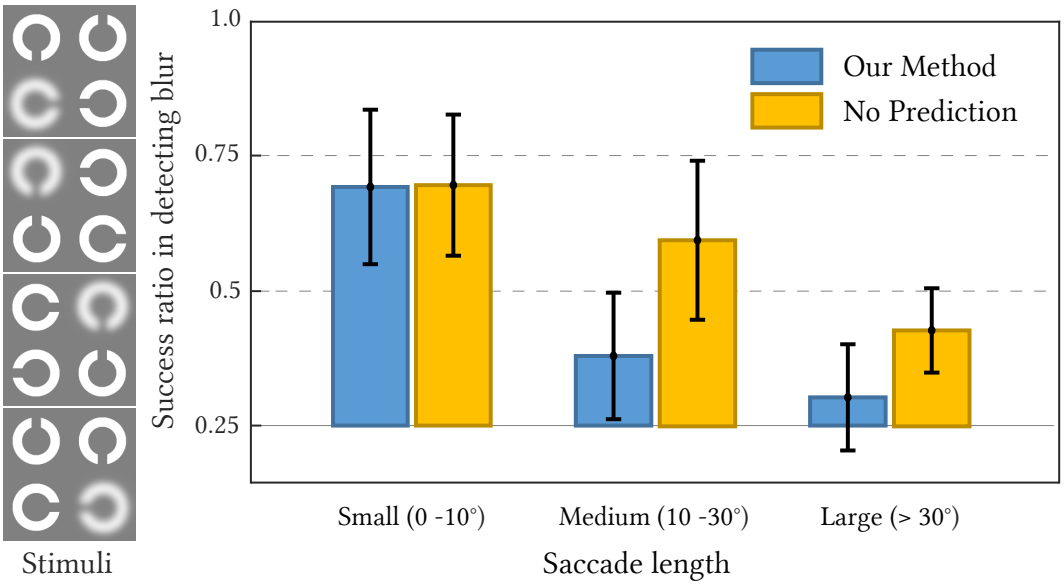


Figure 4.8. Left: Four sample synthetic stimuli, each consisting of four Landolt's C shapes. The size of each stimulus was 1.2°. During the experiment, the orientations and the index of the blurry shape was randomly chosen. The blur was removed when the foveated region moved to the position of the stimulus. Right: Results of the user experiment. A value closer to 0.25 (expected success ratio with purely random choice) is considered more successful at hiding potential artifacts due to the transition from non-foveated to foveated rendering at the positio=n of the stimulus. The error bars correspond to the standard deviation across participants.

2016], the radius of the high-quality rendering region was set to 6.5°, and the render quality of the periphery was reduced by applying a Gaussian filter with the standard deviation of 0.5°. There was a smooth transition between the peripheral and foveal regions which spanned 9°. Using this rendering technique, we conducted a pairwise comparison experiment in two scenarios: one when our prediction was used and the other one without the prediction. In addition to evaluating our prediction using the personalized model, we also measured the prediction performance of the average model. Furthermore, to investigate how the performance changes for systems with different latency characteristics, we considered three different combinations of eye-tracker sampling frequency and display frame rate settings: (120 Hz, 60 FPS), (120 Hz, 30 FPS), and (60 Hz, 30 FPS). As our display could not run at a native 30 FPS, the frame rate was simulated by frame repetition. Figure 4.9 (left) shows 10 images used in this

experiment.

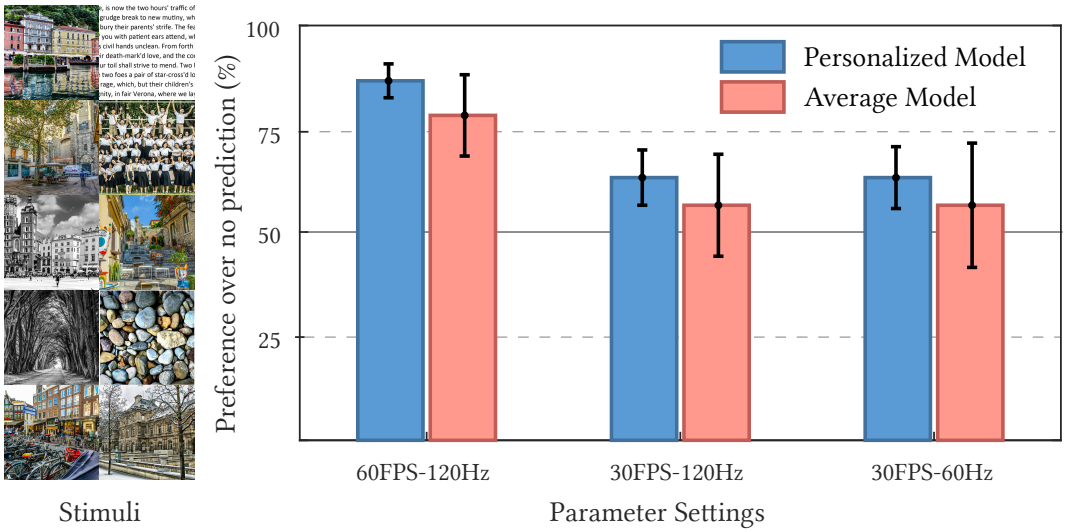


Figure 4.9. Left: Insets of the images used in the free-viewing experiment. The size of each image is 2560 × 1440 pixels. Right: Results of the free-viewing experiment. The error bars correspond to the standard deviation across participants.

Nine participants with normal vision and naïve to the purpose of the test performed the experiment with their personalized models. In each trial, participants were shown an image rendered with our foveated rendering method described above. They could switch freely between the version with and without prediction and investigate the quality without any time constraints. The participants were asked to indicate which version provided higher-quality and more stable images. Each participant investigated all ten images, which took approximately 5 minutes. Images were shown in the native resolution of the display and spanned a visual field of 46° × 27°. The experiment was performed separately for each combination of display frame rate and eye-tracker sampling frequency settings. 6 participants repeated the whole session with the average prediction model, in addition to the personalized model. Participants were allowed to take breaks between sessions or continue the experiment at a different time if they felt fatigue. The eye tracker was re-calibrated at the beginning of each experiment and when the frame and sampling rate settings were changed.

We collected all the responses and analyzed them using a two-sample t-test with a significance level of 95%. Figure 4.9 (right) shows the ratio of cases where the foveated rendering with our prediction was chosen. In all cases, the prediction led to a higher preference. For all scenarios, the improvement provided by

the personalized model over the no-prediction scenario is statistically significant. The personalized model also obtained higher scores than the average model; however, we fail to show a significant difference between the performance of the personalized and the average model ($p > 0.45$). This observation, combined with our previous findings (see Figure 4.7), suggests that the average model is a feasible alternative for the personalized model, even though personalization has the potential to improve the prediction for some of the participants noticeably as shown in Figure 4.7 (right). We also observed that the advantage of using our techniques decreased for setups with higher latency: (120 Hz, 30 FPS), and (60 Hz, 30 FPS). Although the participants still showed a significant preference for rendering with the prediction ($p < 0.03$), the results are not as remarkable as in the case of (120 Hz, 60 FPS). In fact, the difference for (120 Hz, 30 FPS) and (60 Hz, 30 FPS) setups becomes statistically insignificant for the average model ($p > 0.65$). In an informal interview after the experiment, participants frequently reported "tunnel vision" or sudden "pop-up" effects for both ours and standard foveated renderings in the 60 Hz sampling rate and 30 FPS case. Many people suggested that the instabilities are visible in both techniques, and therefore, the difference between them becomes less obvious. This agrees with the fact that for displays with a lower frame rate, e.g., 30 FPS, an image update requires 33.3 ms in the most optimistic case, which is less than the duration of short saccades. This time might therefore be insufficient to provide image updates before the shorter saccades end. This suggests that for the (120 Hz, 60 FPS) setup our technique successfully overcame the problem of latency. For the systems with higher latency, even though we provided an earlier estimation of the next fixation location, this might not have been early enough to overcome the latency problem completely.

Head-Mounted Display We also have experimented with a head-mounted system (Oculus DK2 + Pupil Labs' eye-tracker). We ran a preliminary experiment, similar to the one which was used for the stationary system but with stereoscopic images. Five naïve participants took part in this experiment; they compared standard foveated rendering with a method using our personalized prediction derived with the desktop setup (Section 6.3). We repeated the experiment for 75 FPS and 45 FPS. The sampling rate of the eye tracker was 120 Hz. Each participant performed 28 comparisons in total. The rendering with prediction was preferred in 54 % and 69 % of all comparisons for the first and the second setup. Although these results already suggest that our prediction can be directly applied to head-mounted displays, we encountered several hardware problems which,

we believe, affect the results. First, the quality and the resolution of the screen, especially the blur towards the boundaries of the visual field introduced by the lenses, make the effect of gaze-contingent rendering more subtle. Second, the quality of the data provided by the eye tracker is very sensitive to tiny movements of the headset with respect to the head of the viewer. In particular, small changes in the relative distance between the observer and the screen introduce errors in the conversion from the on-screen location to visual angles. Moreover, we also observed that our eye tracker often loses the gaze direction and takes a significant amount of time to recover. Some of these problems are already addressed in stationary eye trackers, and we believe that this will also be the case for future HMDs as the technology matures. For these reasons, although the initial results are promising, we leave the application of our prediction strategy to head-mounted setups as future work.

4.3 Discussion and Future Work

As demonstrated in the previous section, our technique can provide significant gains when compared to a standard gaze-contingent rendering. In this section, we discuss potential improvements that could be made based on our assumptions by further extending our model.

Within-subject Variability Our data-driven prediction model is based on a ballistic saccade approximation, and for a given pair of time stamp and gaze displacement, it returns a single prediction for the saccade amplitude value. However, it is known that there is a statistical within-subject saccade variability [Leigh and Zee, 2015b, Ch. 3]. For example, the peak velocity of saccades with similar amplitude may depend on the given task, saccade direction, initial and final orbital orientations of the eye, learning, or even on a day-by-day basis [Smeets and Hooge, 2003; Bollen et al., 1993]. The overall good performance of our average prediction model shows that, in practical applications, such a saccade variability may be regarded as another source of noise similar to the measurement noise of the eye tracker. We initially experimented to incorporate such factors into our model, but the amount of improvement was observed to be insignificant. In addition, our performance measurements already included the effects of such deviations from the ideal saccade behavior by having participants in our experiments take part in multiple sessions, at different times of day and with different tasks. While inaccurate predictions cannot be removed online, we make an effort to minimize the effect of the conditions which are known to result in unreliable

predictions. In such cases, we fall back to the standard gaze-contingent rendering. One example of this is raising the detection threshold to $V_d = 180°/s$ to prevent participant-dependent tremor-like eye motions from triggering a false saccade detection.

Between-Subject Variability The elimination of between-subject variability by the personalization of the prediction model leads to a further improvement in the prediction accuracy (Figure 4.7, right). The potential drawback is that the method requires a model-fitting step. At present, we train the personalized models offline. However, our initial experiments indicate that this can be done while running a particular application of gaze-contingent rendering. Such a method could start with our average prediction model and then fine-tune it on the fly using new saccade samples as they are detected. This could also potentially account for some factors that affect within-subject variability, such as fatigue or task-dependent variability. Another advantage of the personalized data-driven model is that it naturally accounts for certain aspects of saccade variability such as corrective glissades [Holmqvist et al., 2011, Ch. 2].

Users with Corrective Glasses The refractive power of corrective glasses bends the light rays on the way from the screen to the eyeball, which affects the measurements of saccade amplitudes when expressed in terms of visual angle. For example, a basic spherical lens that is used to correct nearsightedness (myopia) minifies the world, which effectively means that the time needed to complete the saccade between a pair of points on the screen is shorter for the same person with glasses than without glasses. This obviously affects the precision of our average prediction model, which was obtained only from participants with normal vision. In an informal study we observed that our personalized model could compensate for a medium spherical lens correction, while it failed for more complex progressive glasses with an astigmatic component. We relegate to future work the extension of our personalized model to handle diverse prescriptions of corrective glasses. Use of contact lenses does not affect the performance of our prediction model.

Fixation Prediction and Visual Attention In this work, we focus on predicting the landing position for a single saccade. A significant body of research investigates the saccade planning problem based on the image content and user's task [Kowler, 2011], where an attempt at predicting the saccade sequence is made based on visual attention and "saliency map" modeling [Borji and Itti, 2013; Katti

et al., 2014]. This is a far more complex problem which also involves cognitive and application-dependent aspects, but in principle, the saliency consideration along the saccade trajectory and in the proximity of the predicted landing position could contribute towards an improvement of the prediction accuracy by effectively "snapping" the fixation to the locally most salient feature. We relegate this promising research avenue to future work.

4.4 Conclusion

Gaze-contingent methods promise an improvement in user experience while exploring and interacting with digital content. To provide superior quality, the image updates have to be performed on time according to the current viewer's gaze direction, as any delays may lead to dissatisfaction. In this work, we presented an end-to-end system that uses a saccade landing prediction to combat system latency. The main idea is to take advantage of saccadic suppression and update the image to the new fixation location as soon as the saccade starts. Effectively, such an approach provides an update to the new fixation location before the saccade ends, which leads to less visible delays. To this end, we propose a measurement-driven saccade model that can predict the landing position before the next fixation is established. Our prediction provides better accuracy than existing techniques. Also, an important feature is the continuous prediction refinement as new eye-tracking samples arrive. We applied the model in a simple foveated rendering system and demonstrated significant improvements compared to a system without prediction. A great advantage of our technique is that it comes at almost no additional cost and can be integrated into any existing gaze-contingent system in a straightforward manner. To our knowledge, this is the first work that presents and evaluates a real-time gaze-contingent system with the prediction of saccade landing position.

Chapter 5

Practical Saccade Prediction for Head-Mounted Displays: Towards a Comprehensive Model

5.1 Introduction

System latency poses a serious challenge for gaze-contingent techniques, especially during fast eye movements. Both the lower quality content after the saccade and the late quality change can be often observed by a viewer leading to characteristic popping artifacts. In Chapter 4 we proposed a technique to limit this undesired effect. To perform the quality update of foveated rendering ahead of time, we leverage the saccadic suppression effect, which is the reduced sensitivity of the human visual system during the saccade. To this end, we develop a prediction method that is based on few initial eye-tracker samples to predict the saccade landing position. With the help of such a technique, when the saccade ends, the high-quality foveal rendering is positioned correctly and no popping artifacts are observed.

The success of such a technique is mainly dependent on the accuracy and efficiency of saccade landing position prediction. The method from the previous chapter is computationally efficient, but it relies on the assumption that the saccade displacement profile depends solely on the saccade's length. Other techniques, such as the machine-learning-based approaches by Morales et al. [2018; 2021], define saccades in 2D screen space, but do not account for the fact that saccades are often combined with vergence eye movements. Additionally, the network inference required for the prediction is too expensive to use them in real-time foveated rendering applications.

This work goes beyond existing models for predicting the saccade landing position by investigating additional factors that affect these eye movements and their prediction. More specifically, we focus on dynamic scenarios in VR and AR devices where saccades are combined with vergence movements and smooth pursuit eye motion (SPEM). We design and conduct user experiments that measure saccade profiles in such scenarios. Several previous works, for example [Collewijn et al., 1988b,a], have already conducted similar experiments using accurate eye trackers, such as these using the scleral search coil technique. These studies have demonstrated the impact of the additional factors on the saccade profiles. Compared to them, we do not provide new insights into the physiological characteristic of saccades. Instead, we analyze these factors in the context of practical applications of saccade prediction techniques in VR and AR scenarios. For this reason, we also refrain from using coil-based eye trackers and opt for optical solutions, which, despite their lower accuracy, are the most suitable solution. To our knowledge, this work is the first to investigate the impact of such factors as vergence or SPEM on the saccade prediction in current VR and AR devices. Additionally, we propose a method for correcting existing models for prediction to account for these factors.

We believe that our investigation and technique will also help future developments of machine-learning-based techniques by limiting the required amount of training data.

In this chapter, we present the following contributions:

- analysis of the effects of saccade orientation in 3D space and smooth pursuit eye-motion (SPEM) and how their influence compares to the variability across users,
- a simple, yet efficient post-hoc correction method that adapts existing saccade prediction methods to handle these factors without performing extensive data collection.

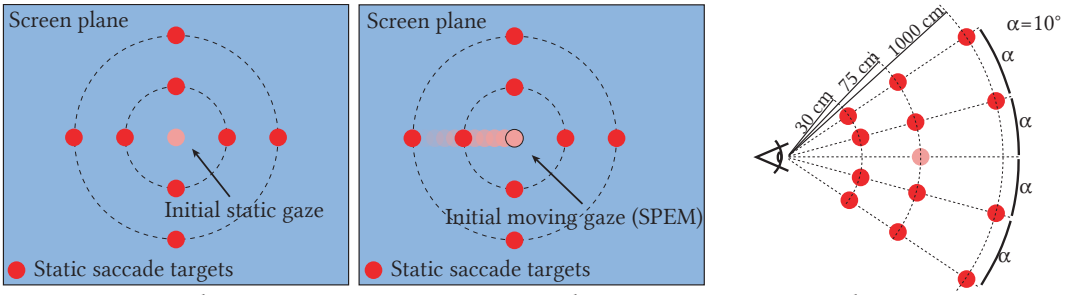
5.2 Overview

This chapter consists of two parts. In the first one, we present a user experiment (Section 5.3) where saccade profiles are collected for different amplitudes, orientations, depth levels, and with and without initial speed. In Section 5.4, we analyze the collected data to discover the most significant factors affecting saccades. In the second part of this chapter (Section 5.5), we present a method

for adjusting existing saccade landing position prediction models to take the analyzed effects into account.

5.3 Experiment design

In our experiment, we aimed to investigate how saccade profiles depend on the saccade's orientation (in 3D space) and initial smooth pursuit eye movements. Additionally, we compared the effects with variability across different users. To this end, instead of conducting separate experiments, each designed to investigate a single factor, we designed the stimuli and the task to simultaneously study all of the effects in different trials of a single experiment. As our main focus are applications of the saccade prediction techniques for head-mounted displays, the experiment was designed for a virtual reality device equipped with an eye tracker.



Static initialization Dynamic initialization Side view

Figure 5.1. The figure presents the main stages of each trial of our experiment. In the initial phase, we had either static initialization (left), where the initial gaze was shown as a static target in the center of the screen, or dynamic initialization (middle), where the initial gaze was moving to stimulate a smooth pursuit eye movement. After 1–2 seconds of the initial phase, the sphere was displaced to stimulate a saccade. Some trials of the experiment included a change in depth to stimulate vergence eye movement as shown on the right.

5.3.1 Stimuli

To guide the eye movements of participants, we rendered a red sphere on a blue background as the visual target (Figure 5.2) at 75cm distance from the virtual camera. In order to preserve the retinal size of the target as 1 visual degree throughout the experiment, the size of the rendered sphere was adjusted depending on its position and distance in 3D space. This prevented the potential

saccadic inaccuracies during the experiment due to the changes in target size when the target displacement involved a change in depth (e.g., fixating on arbitrary parts of the target sphere when it appears bigger at a close distance). Each trial began with an initial phase where the participants are asked to either fixate on a *static target* or follow a *dynamic target*. The duration of the initial phase was randomly selected between 1–2 seconds to avoid anticipation effects.

Static initialization In two-thirds of the trials, a static target appeared at the center of the screen, followed by a target displacement in one of the left, right, up, or down directions to stimulate a saccade between two static positions (Figure 5.1 - left). The displacement amounted to 10° and 20°, respectively for short and long saccades. The trials with a change in vergence involved a simultaneous change in the depth with displacement (to 30cm or 1000cm w.r.t. virtual camera). The target remained visible for 2 seconds at the end of each trial for fully completing the eye movement.

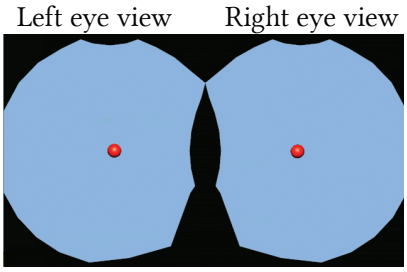


Figure 5.2. The images of the stimuli as shown to the participants of the experiment on the VR display.

Dynamic initialization In the remaining one-thirds of the trials, the target moved along a linear, vertical or horizontal trajectory with a constant velocity of 10°/*s* (motion ramp) to stimulate smooth pursuit eye motion. Motion was followed by target displacement (step) to stimulate a saccade during smooth pursuit eye movement (a.k.a. *ramp-step* paradigm, Figure 5.1 - middle). Target motion started from a source position located on the left/right or above/below the center of the screen for horizontal and vertical trajectories, respectively. The motion was always directed towards the center and it would last for a random duration of 1–2 seconds with the target never exceeding a distance of 10° from the center. Displacement in target step shared similar properties as the trials with a *static target* (i.e., 10° and 20° displacement size with a single final depth of 75cm relative to the virtual camera position).

5.3.2 Task

During the experiment, each participant was asked to fixate on or follow the target with their eyes. The participants could abort the experiment at any time,

especially if they started experiencing viewing discomfort. However, no participant has terminated the experiment prematurely due to viewing discomfort. Each participant was shown the same set of stimuli, but in a randomized order to minimize the bias due to learning effect. The set was constructed according to the cases visualized in Figure 5.1 and contained combinations of:

- 2 orientations of the saccade (horizontal and vertical),
- 2 saccadic amplitudes (10° and 20°),
- 3 depth levels to which the saccade was performed (30cm, 75cm, or 1000cm), and
- with/without initial SPEM.

To keep the experiment procedure simple for the participants, we excluded from our trials the cases where the sphere is moving in the initial phase and is then re-positioned to a different depth. We collected 12 saccades for each of the remaining cases amounting to 384 saccades per participant. The experiment took around 30 minutes to complete. To avoid fatigue, we divided the experiment into 3 sessions with 2 mandatory breaks of at least 10 minutes in between. We had 7 participants (2 of which are authors) with normal or corrected-to-normal vision, ages 25–37, all male. Due to amplified eye tracking inaccuracies associated with the use of eye glasses during a pilot run of our experiments, the participants with corrected-to-normal vision only used contact lenses. Also, to avoid calibration related problems, we introduced an additional verification step after the eye tracker calibration: The users were asked to consecutively fixate on four different targets, also red spheres with visual size of 1°, located at 10° in the periphery in the four primary directions. We repeated the calibration procedure if the estimated gaze location was more than 1° away from any of the four targets.

5.3.3 Hardware

The experiment was implemented using Unity[1] platform and it was ran on HTC Vive Eye Pro headset which provides 1440×1600 px resolution per eye at 90Hz. We used the headset's integrated 120Hz eye tracker which was calibrated at the beginning of each session using the 5-point calibration procedure provided by the eye tracker software. The accuracy of the eye tracker reported by the manufacturer is 0.5°–1.1°, however, recent research [Sipatchin et al., 2021] reports

[1] https://unity.com

different values: 4.16° mean average accuracy of both eyes across field of view of 27° and mean precision of 2.17° for a head-still condition such as our task; the data loss is estimated to be 3.69%.

5.4 Analysis of experimental data

The data from the experiments was used to extract mean saccade profiles which were then analyzed to quantify the influence of different factors. To our knowledge, there is not any common dissimilarity measure for comparing saccadic displacement profiles with each other. Therefore, we also provide a formulation of our measure that helps detecting the most significant factors affecting the saccade.

5.4.1 Saccade profiles extraction

Similar to Chapter 4, our saccade profiles describe the on-screen displacement with respect to the saccade anchor point, as a function of time that elapsed since the beginning of the saccade. To extract the profiles from the data collected in the experiment, we follow the procedure described in Section 4.1.2. We first use a high velocity threshold value for detecting a saccade and then a second, lower velocity threshold value to scan the gaze samples backward in time to find its beginning. The first step gives us the detection point of the saccade and the second - its anchor point at which we assume the saccade has started. This two-step procedure reduces the detection likelihood of false positives and collects the additional samples that are needed to capture the beginning of the saccade. Since we compute the velocity by estimating the distance of consecutive samples without applying a velocity filter, a double threshold policy improves the reliability of correctly detecting saccades.

To analyze the effects of different factors, we define sets of categories belonging to each factor and we classify each saccade of our dataset into one of its categories. Each category contains a subset of the dataset and within the same factor the categories are mutually exclusive. To investigate the influence of orientation of the saccade, we classify the saccades according to the location of their landing position with respect to the initial position of the gaze (factor: ORIENTATIONS, categories: HORIZONTAL, VERTICAL). Similarly, to analyze the influence of depth/vergence change, we classify saccades according to the depth of final position with respect to the initial point (factor: DEPTH, categories: SAME, NEARER, FARTHER). For analyzing the influence of SPEM, we classify the initial eye mo-

tion at the beginning of the saccade which may be performed from a static target, a target moving in the direction of the imminent saccade, and a target moving in the opposite direction of the imminent saccade (factor: INITIAL MOVEMENT, categories: STATIC, SAME, OPPOSITE). Additionally, to analyze differences among subjects we create a category for each person containing only the saccades performed by this individual. (factor: USERS, categories: *each user*). We analyze the factors for short (amplitude 10°) and long (amplitude 20°) saccades separately to verify that observed effects are consistent across amplitudes.

In gaze-contingent rendering applications, inaccuracies in saccade prediction may remain imperceptible if the prediction error is limited. Previous research on the anatomy of the human retina revealed that the angular subtense of the human fovea is approximately 4°–5° [Hendrickson, 2005]. We assume that when the prediction error reaches around approximately half of this distance, the misplacement of foveal region becomes visible by observers. Consequently, an improvement of the prediction error may be evaluated by comparing with this baseline. Therefore, we also included the saccades at a range of 2° difference in amplitude around short and long saccades. More specifically, we consider saccades with amplitudes 9° and 11° for the short saccades, and 19° and 21° for the long saccades in our comparisons (factor: AMPLITUDE, categories: $-1°$, $+1°$). A summary of the factors and the categories we defined for our experiments are shown in Table 5.1.

Table 5.1. The factors that we consider when analyzing saccades and the categories in which we classify them according to each individual factor.

Factors	Categories
ORIENTATION	HORIZONTAL VERTICAL
DEPTH	SAME NEARER FARTHER
INITIAL MOVEMENT	STATIC SAME OPPOSITE
USERS	*Each user*
AMPLITUDE	$-1°$, $+1°$

To analyze the differences within each factor, we aim to compute mean profiles for each category created for it and for each saccade amplitude $\alpha \in \{10°, 20°\}$. For each category, we start by filtering out the saccades that do not belong to it and then align the displacement profiles in the temporal domain. For each factor and each amplitude α, we start by removing all the saccades with amplitude outside the range $[\alpha - 1°, \alpha + 1°]$. In addition, we check the length of the saccades, the direction of SPEM, and the direction of the vergence performed by the participants to label the samples that

do not conform with the expected behavior in the category as outliers. Then we align the anchor points of saccades by applying a temporal offset (Section 4.1.2). In our case, we choose the velocity threshold for detecting a saccade as 180°/*s* and the anchor point as 90°/*s*. All eye-tracker samples 30 *ms* prior to the anchor point are included in the analysis as well. To obtain the mean profile sampled at equal time intervals, we resample each profile using linear interpolation of measured displacements. Our eye tracker operates at 120Hz and provides a gaze estimations every 8 *ms* or 9 *ms* and not all of these samples are valid. Therefore, the samples for the different saccades are at different time positions with respect to the their beginnings. Resampling at equal intervals is needed to align the displacement values for each saccade in the same time positions. We chose our interval to be 1 *ms*. Since the endpoint of each saccade occurs at an arbitrary time, we consider the endpoint of the mean profile to be positioned at the mean time position of all the endpoints.

After the above initial processing, the mean profiles are computed by averaging samples of all saccades within each category. Formally, we represent those mean profiles as a sequence of N mean samples computed from the original profiles:

$$\overline{S} = \{\overline{s_0}, \overline{s_1}, ..., \overline{s_N}\}, \tag{5.1}$$

where each sample $\overline{s_l} = (t_l, \overline{d_l}, \sigma_l)$ is defined by its timestamp t_l, mean displacement $\overline{d_l}$, and the standard deviation of all the displacement values for the given timestamp σ_l within the category. The first sample of the saccade ($\overline{s_0}$) is the anchor point ($t_0 = 0$) whereas the last sample ($\overline{s_N}$) is the end point ($\overline{d_N}$) and it is equal to the amplitude of the saccade. Figure 5.4 visualizes the mean displacement profiles for all categories grouped by the factors they belong to.

5.4.2 Dissimilarity measure for saccade displacement profiles

To be able to analyze and compare the effects of different factors, we propose a dissimilarity measure for quantifying the differences between the mean saccade profiles corresponding to individual categories within a single factor. More precisely, for a given set of mean profiles $\{\overline{S^k} | \overline{S^k} = \{\overline{s_0^k}, \overline{s_1^k}, ..., \overline{s_{N_k}^k}\}\}$ belonging to the categories within a factor (Table 5.1), we define a measure that correlates with the differences for that factor as:

$$D(\{\overline{S^k}\}) = \sum_{l=0}^{\min_k N_k} \frac{\max_k \overline{d_l^k} - \min_k \overline{d_l^k}}{\max_k \sigma_l^k}, \tag{5.2}$$

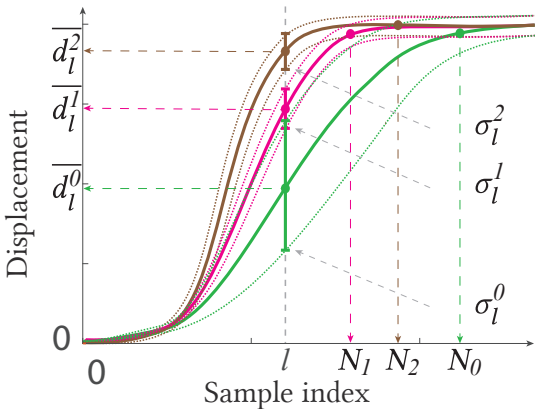


Figure 5.3. Mean displacements ($\overline{d_l^k}$) and standard deviations (σ_l^k) that we used in Equation 5.2 are shown for hypothetical mean saccade profiles. Mean displacement profiles of three categories; namely, $\overline{S^0}$, $\overline{S^1}$, and $\overline{S^2}$, are represented by green, pink, and brown solid lines, respectively, whereas whiskers and dotted lines visualize the standard deviation of saccade displacements from the corresponding category.

where k is the index of a category. The measure can be seen as an area between the upper and lower envelope of all mean displacement profiles for the factor ($\{\overline{S^k}\}$), normalized by the maximum standard deviation of the displacement values observed for the factor ($\max_k \sigma_l^k$). Figure 5.3 illustrates an abstract example of the mean displacements ($\overline{d_l^k}$) and the standard deviations (σ_l^k) of three hypothetical sample categories. It is important to note that this measure requires the mean displacement profiles to have equal sampling intervals.

Computing the dissimilarity measure in Equation 5.2 yields a higher value if there is a more significant difference between the mean saccade displacement profiles corresponding to different categories within a given factor (Table 5.1). The differences between the categories commonly manifest themselves through speed ups or slow downs in saccade displacement profiles and we use our dissimilarity measure for identifying perceptually significant changes to the displacement profiles that require an update to the prediction model or training data to avoid visual artifacts.

5.4.3 Discussion

Figure 5.4 summarizes the effects that different factors have on the mean displacement profiles. Additionally, we provide a bar plot of our dissimilarity measure for each factor.

A clear difference can be observed for the case where we compare mean displacement profiles with different amplitudes (factor: AMPLITUDE). The difference for both 10° and 20° saccades is consistent with the fact that longer sac-

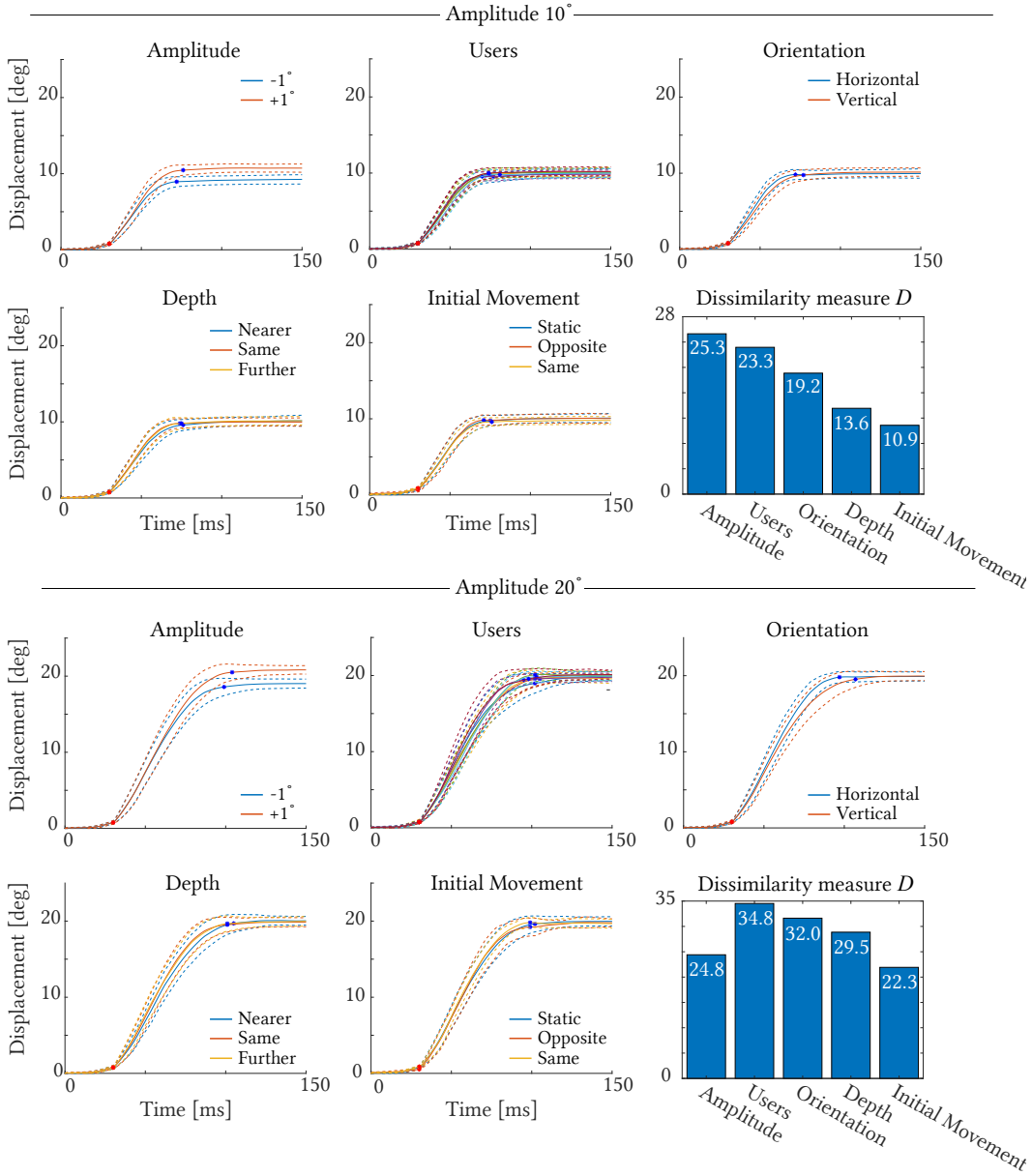


Figure 5.4. Effects of different factors on the saccade mean displacement profiles computed from our experiment data. The solid lines represent mean profiles for each category, while the dashed lines visualize corresponding standard deviations. The bar plots show the values of our profile dissimilarity measure (Equation 5.2) for different factors (Section 5.4.1). The bars representing the AMPLITUDE factor are provided as a reference baseline for the minimum value of similarity to observe a significant effect (please refer to Section 5.4.1 for details).

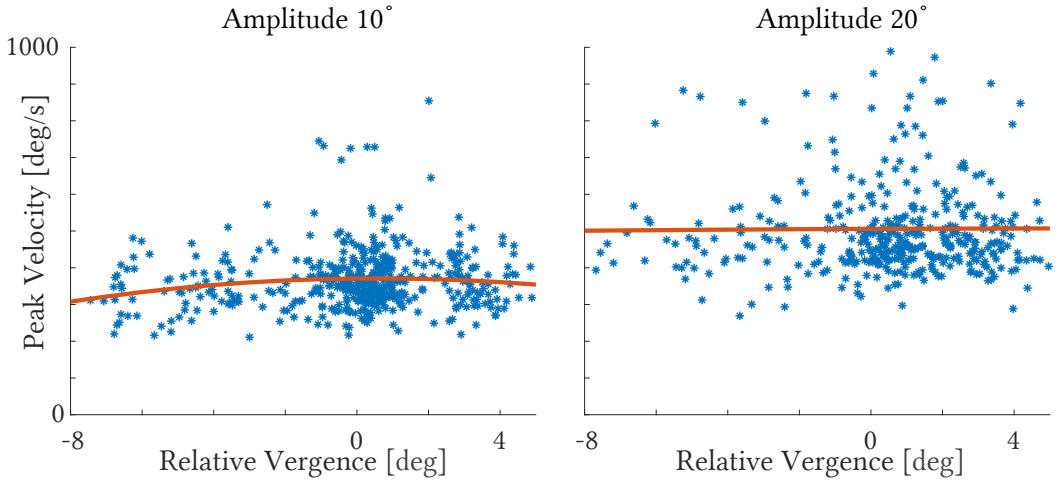


Figure 5.5. The relation between the vergence change during the saccade and the peak velocity. The blue points correspond to individual saccades, while red curves are the quadratic line fits showing the overall trend. Negative values indicate saccades that move closer to the observer, whereas positive values indicate saccades moving further away.

cades exhibit steeper ascend in their displacement profiles compared to shorter saccades. Existing saccade landing prediction models depend on these profiles to be distinguishable, which is an expected effect. It also serves as the baseline to compare the difference exhibited by the other factors of interest as we mentioned in Section 5.4.1. Therefore, we aim to identify the effects that will change the performance of saccade landing position prediction and assume that factors that provide smaller effects than what is observed at 2° change in the saccade amplitude may not lead to significant improvements in applications that rely on the prediction. In particular, the value of the dissimilarity measure D, 25.3 for 10°, and 24.8 for 20° saccade are the reference points for analyzing the effects of the other factors.

Apart from the AMPLITUDE factor, the most significant differences were observed for USERS. The differences become even more apparent for longer saccades (20°). It has been already shown in Chapter 4 that tailoring a model to fit the personal saccadic characteristics of a user leads to lower saccade prediction error and to a higher subjective preference for that user compared to the model trained for the average population. While they demonstrated this in a task-performance experiment, here we demonstrate the underlying difference in saccade profiles.

The third factor with the highest differences was ORIENTATION. Similar to

USERS, the differences for 10° saccades were smaller than for the AMPLITUDE factor, but the opposite can be observed for 20° saccades. For this factor, the differences become close to those observed with the AMPLITUDE factor.

Contrary to our expectations, moving the target to different depth levels (factor DEPTH) led to smaller changes in the mean displacement profiles, especially for 10° saccades. While we observed some changes in the peak velocity (Figure 5.5), the differences are smaller than those reported by the previous studies (Section 2.2.4). We relate this discrepancy with the existing studies mainly to the profound difference between the real and virtual environment. First, standard head-mounted displays are not fully capable of reproducing accommodative cues, and any depth change only results in a change in the vergence (due to the change in disparity), but it does not trigger an accommodation response from the participants' visual system. The lack of an accommodation response may be seen as a deviation from real-world viewing conditions, but it applies to most mainstream stereoscopic HMDs used for virtual reality. Therefore, we have not tried to mitigate this effect in our experiments. Second, similar to all experiments conducted on stereoscopic displays with a lack of accommodation response, the presence of well-known vengeance-accommodation conflict [Shibata et al., 2011] imposed a limit on the depth ranges that we could test in our experiments without causing viewing discomfort for the participants. These differences between virtual reality and real-world viewing conditions may explain the discrepancy between our measurements and the previous studies, most of which are conducted under real-world viewing conditions. Additionally, the choice of the stimuli could affect the outcome of our experiments. While the small spheres used in the experiment enable precise control over the participant's gaze location and saccades, the fact that they do not change their size according to the distance removes the size cue. The lack of this cue could potentially influence the saccade accuracy. Also, the use of specific colors, red and blue in our case, may lead to a different amount of edge blur due to the wavelength-dependent accommodation.

With the final factor, INITIAL MOVEMENT, we observed that the initially moving target led to smaller differences in the displacement profiles. We believe that higher pursuit speeds could potentially enhance the effect. In our experiment, we chose a moderate pursuit speed (10°/s) to keep the task simple and give the observer ample time to properly fixate on the moving target and initiate SPEM. Similar to DEPTH, the differences become larger for 20° saccades, and they are close to these observed with the AMPLITUDE factor. It is possible that the differences become more apparent for more extreme saccade amplitudes. Unfortunately, reliable measurement of larger amplitude saccades poses problems

due to the fact that virtual reality headsets have a limited field of view with high fidelity.

In all our experiments, we used an optical eye-tracker, the current technology of choice for VR and AR applications. Despite its widespread use, this technology is not suitable for capturing all the characteristics of eye movements [Nyström et al., 2013; Hooge et al., 2015; Nyström et al., 2016; Hooge et al., 2016]. In particular, due to post-saccadic oscillations of the pupil, the optical eye trackers have low accuracy in estimating the saccade onset, peak velocity, and its end. Additionally, the sensitivity of the eye trackers to the changes in the pupil's size [Drewes et al., 2014; Jaschinski, 2016; Hooge et al., 2019] has a detrimental effect on the correct estimation of the vergence and the binocular fixation point. To address these limitations and measure eye movements more accurately it is possible to use eye-tracking technology such as the wearable scleral coil tracking system proposed by Whitmire et al. [2016]. However, most users might find coils to be a very invasive way to track their gaze orientation, and to our knowledge, no commercial VR or AR headset uses such technology. Therefore, in our work, we focus on optical eye tracker technology, which, despite its limitations, has been already shown to be beneficial in applications such as foveated rendering [Guenter et al., 2012; Patney et al., 2016]. At the same time, it is important to note that the generalization of our findings to of scleral coil eye-tracking technology needs further investigation.

In the remaining part of the chapter, we demonstrate a new technique that accounts for differences in the saccade profiles to provide better saccade prediction. For demonstration purposes, we chose to focus on two factors that exhibit the highest differences, i.e., USERS and ORIENTATION. While the personalization of the prediction model for a specific user was demonstrated in Chapter 4, the process required collecting a large set of saccades. Here, our goal is to reduce the amount of required data. On the other hand, to our knowledge, adjusting existing models to adapt to the saccade's orientation has not been done before, but our findings suggest that it could improve the prediction accuracy. Therefore, designing prediction methods or adjusting the existing ones to handle different orientations correctly may provide additional benefits in the final applications.

5.5 Method for tuning saccade prediction models

In Section 5.4, we analyzed how different factors affect the displacement profiles of the saccades. We observed the dissimilarity for the individual factors to be comparable to the dissimilarity for the AMPLITUDE factor, with the biggest

ones for USERS and ORIENTATION factors. The observed dissimilarities suggests that incorporating factors such as saccade orientation or the difference among users may improve the saccade prediction. However, the fundamental problem in deriving a model which captures such dependencies lies in data collection. Individual saccades collected for training such models contain noise; therefore, many of them have to be combined to create a reliable prediction. For example, the prediction model in Chapter 4 requires each participant to perform 300 saccades. Still, the model does not capture factors other than the saccade amplitude. Consideration of additional factors, such as orientation, depth, and SPEM, would significantly increase the number of the required saccade samples, making the data collection for individual users tedious and sometimes infeasible. Similarly, most machine-learning approaches, such as Morales et al. [2018], have high data demands for training.

To address the problem of data collection, we propose an alternative approach. Instead of exhaustively collecting data from psychophysical experiments, which enables training prediction models to capture all factors, we postulate that the influence of many factors, such as orientation or user, can be approximated by a low-parameter transformation of the data. The advantage of such a solution is that the effect of additional factors is captured using a small number of parameters, and therefore, such a model is more robust to noise and the reduced number of collected saccades. Successful applications of this approach are shown in the past, such as the method of Lesmes et al. [2010], which uses the a priori information about the contrast sensitivity function's (CSF) general functional form to maximize the information gained from a small number of measurements. Similarly, in this work, we seek a global transformation of the profiles of a saccade prediction model, which has a small number of parameters, yet allows for explaining the effects of additional factors influencing the saccade performance.

The main observation behind our solution is that the differences in the saccade profiles can be attributed to the changes in the saccades' performance/velocity caused by the factors that we investigated in our experiments. This observation can be made by looking at the differences among slopes of the individual mean saccades profiles in Figure 5.4. We demonstrate that these changes can be effectively modeled by shearing the profiles parallel to axis representing the time domain (Figure 5.6). Additionally, we observe that the appropriate shearing factor changes with saccade's amplitude, but we show that this change can be approximated with a low-degree polynomial. This is the key to our technique, as it allows us to compute the shear factor for few saccade amplitudes and then interpolate or extrapolate the shearing transform to the other amplitudes. Below,

we provide a formal definition of shearing (Section 5.5.1) and a shear between two saccade profiles (Section 5.5.2). Then, we describe the derivation of the shearing-based transformation of saccade profiles and how it can be applied to modify a prediction model to account for additional factors in Section 5.5.3.

5.5.1 Shearing saccade profiles

Given a saccade profile $S = \{s_0, s_1, ..., s_N\}$, where each sample is defined by a couple of scalars $s_l = (t_l, d_l)$ representing time stamp, t_l, and corresponding displacement, d_l, we define a sheared version of the profile by applying a 2D shearing parallel to the time axis followed by resampling to restore uniform sampling in time domain. More formally, to shear the profile S with a shearing factor λ, we first transform its samples using a 2D shearing matrix:

$$\begin{bmatrix} \widehat{t_l} \\ \widehat{d_l} \end{bmatrix} = \begin{bmatrix} 1 & \lambda \\ 0 & 1 \end{bmatrix} \begin{bmatrix} t_l \\ d_l \end{bmatrix}, \quad \lambda \in [-1; 1]. \tag{5.3}$$

The resulting profile $\widehat{S} = \{(\widehat{t_1}, \widehat{d_1}), (\widehat{t_2}, \widehat{d_2}), ..., (\widehat{t_N}, \widehat{d_N})\}$ is not sampled regularly at 1 ms intervals anymore after applying the shearing transformation because the time stamp, t_l, of each sample changes. Therefore, we apply a simple linear interpolation to resample it back to 1ms intervals and obtain the final sheared profile. In the rest of the chapter, we denote shearing as a function Ψ, and a saccade or mean saccade profile S sheared with shearing factor λ as $\Psi(S, \lambda)$.

5.5.2 Computation of shearing transformation between saccadic profiles

Given two saccade profiles $S^k = \{s_0^k, s_1^k, ..., s_N^k\}$, where $k \in \{1, 2\}$ and $s_l^k = (t_l^k, d_l^k)$, we can compute a shearing factor λ that describes the difference between those two profiles. Formally, we define the shearing between S^1 (original profile) and S^2 (target profile) as $\lambda = \Lambda(S^1, S^2)$ for which the 2D shear applied to S^1 minimizes the difference with respect to S^2, i.e.,:

$$\Lambda(S^1, S^2) = \underset{\lambda}{\operatorname{argmin}} \sum_{l=0}^{N} |d_l^* - d_l^2|, \quad \text{subject to } S^* = \Psi(S^1, \lambda). \tag{5.4}$$

This definition relies on same sampling of time domain by all profiles involved in the computation (S^1, S^2, S^*). This is, however, guaranteed by the definition of Ψ. The above minimization problem can be easily solved using binary search. Figure 5.6 demonstrates two examples of how the shear between two saccade profiles can be used to align them.

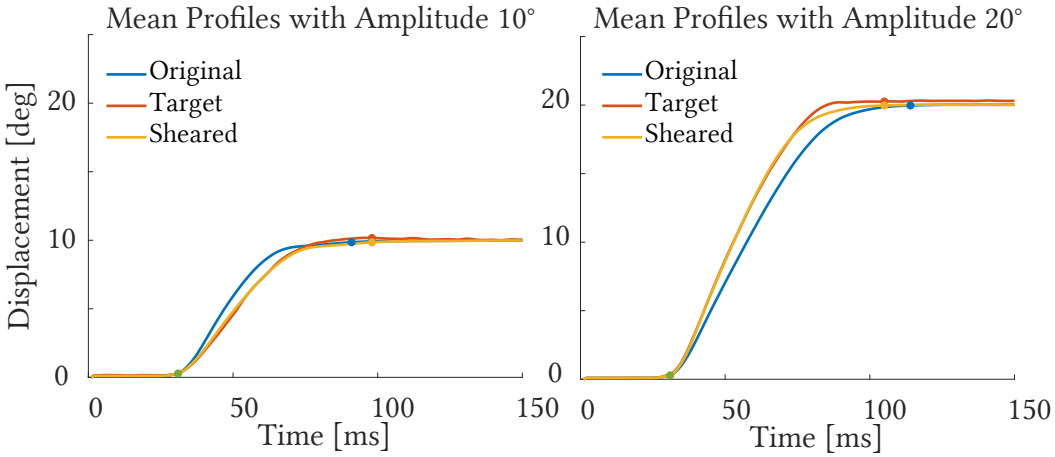


Figure 5.6. Two examples of shearing original mean saccade profiles to match different targets. The target on the left represents a category with slower saccades than the original. The target on the right represents a category with faster saccades.

5.5.3 Application

Previous models for predicting saccades, such as the one from Chapter 4 and the two proposed by Morales et al. [2018; 2021], are trained on large datasets containing saccades with various amplitudes and orientations collected from multiple users using an eye tracker. These models do not account for all the factors analyzed in Section 5.4. Here, we demonstrate how to use the shearing strategy described in Section 5.5.1 and 5.5.2 to account for these factors. It is possible to apply the shearing transformation to saccade displacement profiles directly if the dataset is available (*data shear*). In some cases, although the model is accessible, the dataset that the models were trained on may not be available. For such cases, if possible to extract saccade displacement profile approximations from the model, we apply the transformation to the recovered profiles instead (*model shear*).

Data shear The first approach we consider that utilizes the idea of shearing the saccade profiles is to transform all the saccades in the dataset to create a new dataset that represents a particular type of saccades and then recompute the prediction model using the augmented dataset instead of the original one. In particular, we consider here shearing the saccade profiles to create a dataset and models for horizontal, vertical, and personalized saccades. In all three cases, we apply our shearing strategy in the same way. First, to obtain the specific saccades

for each category (horizontal, vertical, or a particular user), we extract the corresponding saccades from the dataset. These saccades act as a target for the required shear computation applied to the remaining saccades to compute the final dataset. We then discretize the amplitude domain. In our experiments, we chose the discretization step to be 1°. For each discrete amplitude value α, we estimate the mean displacement profile by averaging the saccade displacement profiles with amplitudes in the range $\{\alpha-1;\alpha+1\}$ for both the original dataset and for the target dataset following the same procedure as described in Section 5.4.1. Here, we denote the mean profiles of amplitude α in the original dataset as $\overline{S_o^\alpha}$ and the target dataset as $\overline{S_t^\alpha}$. The number of saccades constructing $\overline{S_t^\alpha}$ we denote with $|S_t^\alpha|$. Note that $\overline{S_o^\alpha}$ and $\overline{S_t^\alpha}$ are mean saccade profiles of the same amplitude. The only difference is that $\overline{S_o^\alpha}$ comes from the original dataset, which contains all types of saccades (e.g., all orientations) while $\overline{S_t^\alpha}$ is a mean profile for the specific category (e.g., horizontal, vertical, or for a particular user). The goal is to use this correspondence to define the shear that needs to be applied to the original dataset, to make it represent a particular category of saccades. To this end, we compute a series of shearing factors $\{\lambda_{\alpha_0},\lambda_{\alpha_2},...,\lambda_{\alpha_M}\}$ for each amplitude α_i following Equation 5.4, i.e., $\lambda_{\alpha_i}=\Lambda(\overline{S_o^{\alpha_i}},\overline{S_t^{\alpha_i}})$.

The main objective of such a dataset derivation is to obtain a large dataset of saccades while using only few measured profiles. To this end, we propose to first collect a subset with a particular category of saccades and compute the shear (Section 5.5.2) of the mean profiles with respect to the mean profiles in the large dataset. Using this procedure, we obtain the relationship between the profiles in the large dataset and newly collected one for a few saccade amplitudes. To obtain the shearing factors for the whole range of saccadic amplitudes, we use a linear regression to fit the linear function $f(\alpha)$ that minimizes:

$$\sum_{i=1}^{M}\frac{\left|f(\alpha_i)-\lambda_{\alpha_i}\right|}{|S_t^\alpha|}, \tag{5.5}$$

where $|S_t^\alpha|$ is a weighting argument used to balance the data in the cases when different amplitudes are unequally represented in the target dataset. This function allows us to estimate the shear required for transforming each profile in the large dataset based on the saccadic amplitude (Figure 5.7). Having the shearing factor for each amplitude α, we apply shear $f(\alpha)$ to all individual profiles to form the new dataset. We compute such datasets for horizontal and vertical saccades, as well as for each user separately.

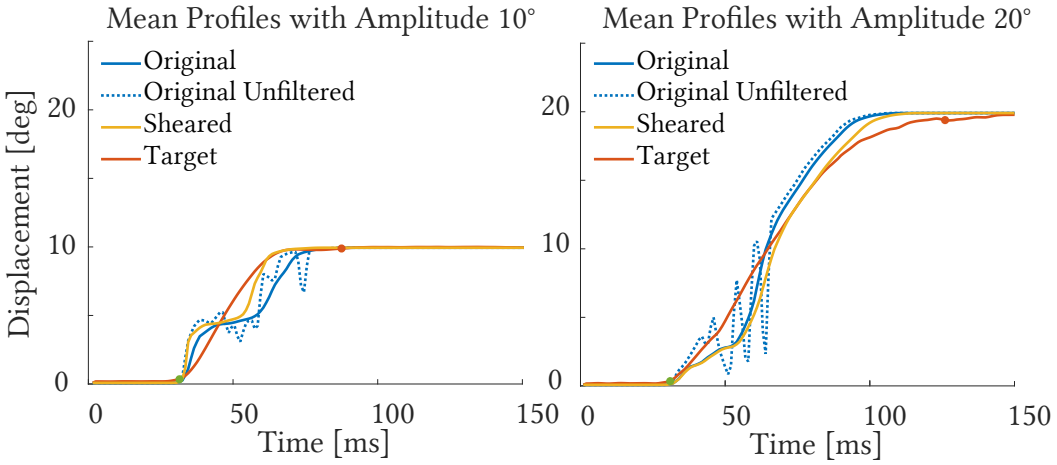


Figure 5.8. Two examples of shearing saccade profiles, recovered from a model, to match the mean target profiles. The target on the left represents a category with faster saccades than the original recovered profile. The target on the right represents a category with slower saccades.

Model shear It is possible to apply the shearing operation directly to an existing saccade prediction model as long as the individual saccade or mean saccade profiles can be recovered. Example of such a model is the one described in Chapter 4. The model provides a mapping from time and displacement pair (t, d) to the predicted saccade amplitude α. Since the model is represented directly by the (t_i, d_i, α_i) triplets, the mapping can be inverted by fixing α_i and treating the corresponding (t_i, d_i) sequence as a displacement profile for a saccade with α_i degree amplitude. Because the model is represented by a discrete number of (t_i, d_i, α_i) sample points, we propose to use a linear interpolation on the displacement values to obtain saccade profiles sampled at regular, one-millisecond, intervals. The blue dotted lines in Figure 5.8 show examples of the displacement profiles obtained using this procedure. Unfortunately, the profiles are often noisy, which prohibits a di-

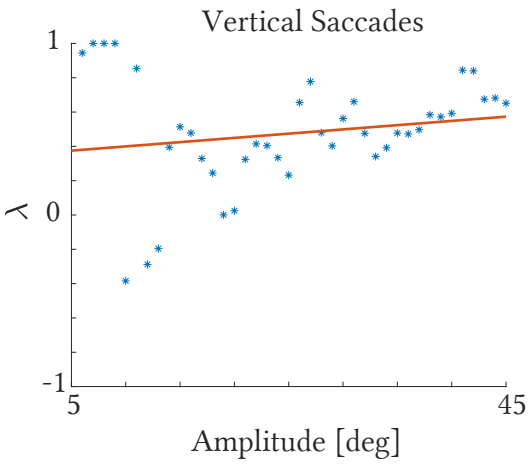


Figure 5.7. The linear function $f(\alpha)$ (red) is fitted to the shearing factors λ_{α_i} (blue).

rect application of the shear with a satisfactory performance. For this reason, as well as to prevent the occurrence of any aliasing, before shearing, we denoise the profiles by first applying a median filter with a window size of 15 ms followed by a Gaussian filter with a window size of 5 ms for smoothing (Figure 5.8, blue solid lines). The values of the window size were chosen heuristically as the smallest values producing stable results. After shearing the individual saccade profiles (Figure 5.8, yellow lines), the new triplets $(\hat{t}_i, d_i, \alpha_i)$ can be used to create a new model. Note that the shearing operation affects only timestamps, and the other components of the triplets do not change. In the particular case of our model from Chapter 4, it is enough to resample the data to be uniformly sampled in time and displacement domain. We, therefore, apply linear interpolation to obtain $(t_i, d_i, \hat{\alpha}_i)$ triplets, where t_i and d_i are sampled at the intervals of the original model, and $\hat{\alpha}_i$ is the new prediction of the saccade amplitude.

5.5.4 Results

In our analysis, we consider both *data shear* and *model shear* strategies described in Section 5.5.3 to update saccade datasets and prediction models. We analyze the effectiveness of these strategies in two different experiments. In the first one, we show an application of shearing operation to update the existing dataset and prediction models for improved predictions when saccade orientation changes (horizontal vs. vertical). In the second experiment, we demonstrate the application of shearing operation to create user-specific models, aka personalization.

We compute our results on the saccade dataset and model from Chapter 4, which includes 6600 saccade profiles collected from 22 participants (300 saccades for each participant). The amplitudes of saccades are evenly distributed in the range of $5° - 45°$. To customize the models for vertical and horizontal saccades, we classify saccades into horizontal and vertical categories depending on their orientation (with +/- 15 degrees allowance around the corresponding orientation). It is important to mention that, while the amplitude distribution across participants is balanced due to the experiment design, this is not the case for the orientation. Due to the aspect ratio of the screen (16:9), the amplitudes of vertical saccades are limited to the range of $5°-22°$ while horizontal saccades have amplitudes up to 40°. Moreover, horizontal saccades are more frequently represented in the dataset, constituting 30% of the collected data, compared to 5% for the vertical saccades.

The baseline for all of our comparisons consists of two models. The first one is the *average model* from Chapter 4. It is derived from our dataset and is based on the interpolation of the collected data. We include this model in our compar-

Table 5.2. Short descriptions of the four models that we compare in Section 5.5.4. Each model is created following the procedure described in Chapter 4, either using our full dataset or a subset of it that includes a single category of saccades (Table 5.1). For data shear we modify the dataset before creating the model and for model shear we first create the model and then modify it to match a specific subset.

Model	Original dataset	Target dataset	Model description
Average	Full	-	The model is created using the original full dataset.
Model Shear	Full	Subset	The model is first created using the original full dataset, and then modified to match a specific subset of it.
Data Shear	Full	Subset	The model is created from an augmented full dataset, modified to match a specific subset of the original dataset.
Customized	Subset	-	The model is created from a specific subset of the original dataset.

isons because it provides a good balance between accuracy, performance, and data volume requirements. However, it accounts only for the variance in saccade profiles due to changes in the amplitude and it does not account for any additional factors that we considered in this chapter (Section 5.4). The second model is the so-called *customized model*, which is derived following the computation of the *average model*, but using a subset of the data corresponding to a specific category of saccades (e.g., for horizontal or vertical saccades). Table 5.2 gives a short summary of the four models that we compare in this section.

In the first experiment, we computed the *customized model* for the two categories of orientation (horizontal and vertical) separately. The number of saccades in each category was sufficient to properly train these models. Later, we used *data shear* and *model shear* as described in Section 5.5.3 to compute two alternative models and compare them with the *customized models*. For *data shear*, we sheared the displacement profiles of all saccades from the dataset, irrespectively of their orientation, according to the shear factor computed by using preselected horizontal and vertical saccades as target. For *model shear*, the shearing factors were computed based on the comparison of the original model from Chapter 4 and the subsets of vertical and horizontal saccades. As for the second experiment,

we followed a similar procedure to evaluate the performance of the shearing operation for personalizing the models, but in that case, saccades of a particular user were selected to compute the shearing factors.

Figure 5.9 presents the performance of different models tailored to the orientation of the saccade. The figure presents both the mean absolute error (left), as well as the mean absolute error for predictions made at a specific moment during the saccades (right). The performance of the horizontally oriented *data*

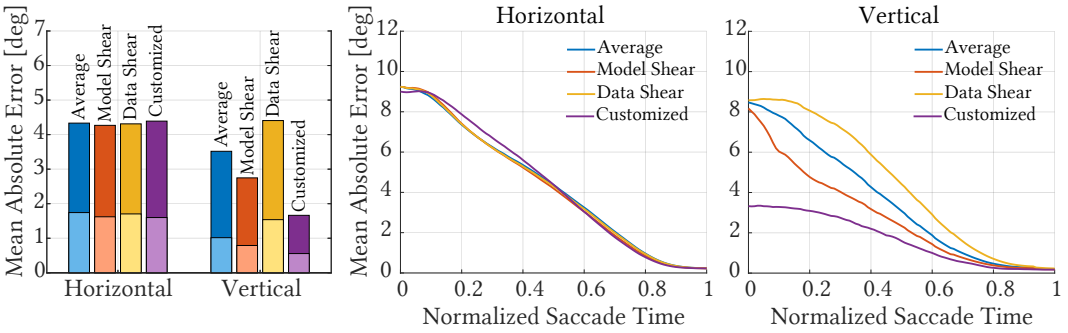


Figure 5.9. The figure presents the performance of differently derived models for horizontal and vertical saccades. The left-most plot presents aggregate mean errors for predictions made for the entire saccade duration (height of the bar) and for the second half of the duration (light segment). The two other plots present the error as a function of duration of the saccade, i.e., at which stage of the saccade the predictions was performed. While the customized model performs best, the sheared model, which requires significantly lower number of saccades for training, performs better than the average model, which does not account for the orientation of the saccade.

shear and *model shear* models is indistinguishable from the *average* one. We attribute the lack of an effect to the predominance of the horizontal saccades in the dataset, and consequently, better prediction of these saccades. In comparison, the prediction for vertically oriented saccades greatly benefits from a vertically oriented models. It is important to mention here that the *customized model* greatly benefits from the significantly lower range of amplitudes in the set of vertical saccades. More precisely, the range of the horizontal saccades is double the one of the vertical saccades due to the dimensions of the display used for the data collection (Section 4.1.1). While reducing the training and testing range of saccades' amplitudes improves the prediction as the error is bound to this range, the model is limited to shorter saccades. In contrast, the models derived using *model shear* and *data shear* support the larger range of amplitudes represented

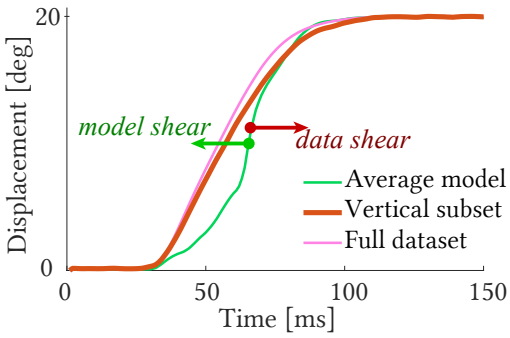


Figure 5.10. Comparison of the mean saccade profiles from the full dataset (pink) and from the vertical subset (orange) with a recovered one from the average model (green). The difference between the two mean profiles indicates slower performance of the vertical saccades with respect to the rest but the sample profile recovered from the average model does not reflect it.

in the original dataset.

While both *model shear* and *customized model* provided a better performance for the vertical saccades than the *average model*, surprisingly, the *data shear* did not improve the model. To understand the reason behind it, we analyzed the mean saccades profiles from the full dataset and from the vertical subset, as well as the cross-section of the original average model in Figure 5.10. When comparing the the mean saccade profiles representing the full dataset and the vertical subset, the first profile requires shearing to the right to match the second. This is expected as the vertical saccades are slower (Section 5.4.3). However, the cross-section of the model exhibits the opposite behavior, i.e. it requires shearing to the left to match the vertical saccades profiles. When applying the *model shear*, the shearing computed based on the vertical saccade profiles and the model results in the model shearing to the left (green arrow), hence, better aligning with the vertical saccades and reducing the error. However, shearing all the profiles in the dataset according to the difference between their representative mean profile and the vertical mean profile, i.e., *data shear*, leads to a sub-optimal shear of the model to the right (red arrow), hence, increasing the prediction error, i.e., worse alignment with the vertical saccades profile. This demonstrates that although *data shear* can perform a correct transformation to the individual profiles, it cannot account for the built-in biases in the model. In this case, this leads to a lack of improvement when *data shear* is followed by the model computation. Conversely, the *model shear*, which computes the shearing factor based on the model, can account for biases in the model and improve the prediction.

The great potential of our shearing strategy lies in the fact that it may allow for training models using significantly lower number of samples than it is required for training *customized models*. To verify this, we analyzed the performance of our shearing strategy for different numbers of saccades (Figure 5.11).

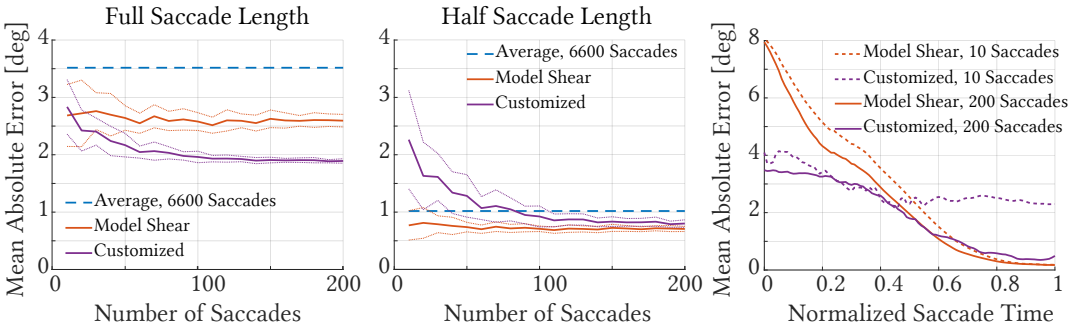


Figure 5.11. Performance comparison for different models as a function of the number of saccades used for their computation. While the plot on the left shows the average error of the prediction for the full length of the saccade, the center plot shows the error for the prediction during the second half of the duration. The solid lines are the means computed using bootstrapping with 20 repetitions, the dotted lines are the corresponding standard deviations. The plot on the right compares the average error of the prediction at any point during the saccade when using 10 and 200 saccades for training the models.

To this end, we divided the dataset of vertical saccades into training and testing sets which consist of 200 and 150 saccades, respectively. By considering different number of saccades (x-axis in the plot) from the training set for computing the shearing factor and the new model, we analyzed the resulting mean absolute error of the prediction. We compared this *model shear* strategy, to the straightforward computation of the model based on the smaller number of training saccades (*customized model*). As expected, when the number of considered saccades is large, the improvement from our shearing technique may be limited. However, we can achieve a better prediction performance, in the presence of significantly lower number of saccades. This is particularly visible for the prediction in the second half of saccade duration, which is critical for techniques such as foveated rendering, where the sensitivity of the visual system is gradually restored towards the end of the saccade (Section 2.2.4). This can be in particularly observed in the right plot in Figure 5.11, where the error is analyzed for predictions made at different points of the saccades' duration. It can be observed that the *customized model* trained on a low number of saccades retains the high error throughout the entire duration of the saccades. In contrast, the error for the model trained using our method drops significantly towards the end of the saccades.

In Figure 5.12, we provide the mean absolute error of predictions obtained from different models for personalization. We observe that for many participants (e.g., users 4, 12, 15, and 21) the prediction performance of models follow an

expected pattern, where *customized model* has the best performance due to the availability of full data used to calibrate such a model. *Data shear* and *model shear* provides the best prediction performances after *customized model* and they are suitable for improving existing dataset or model prediction performances without large data collection requirements for personalization. The *average model* performs least favorably due to the lack of user-based adjustments in saccade displacement profiles. Nevertheless, using a limited dataset for training prediction models is more prone the noise inherent to data. We observe that for some of the participants (e.g., users 7, 8, and 14) *model shear* performs more favorably than *data shear* and we attribute this observation to the model adjustments in *model shear* that are more robust against noise. In some of the cases (e.g., users 3, 18, 19, and 20), *data shear* and *model shear* have a performance level close to that of the *average model*. We believe that for those users, the personalization does not offer a high level of improvement in the performance. However, we observe that the general behavior of mean absolute errors favors the use of *data shear* and *model shear* for improving prediction performance without the cost of collecting a large amount of training data.

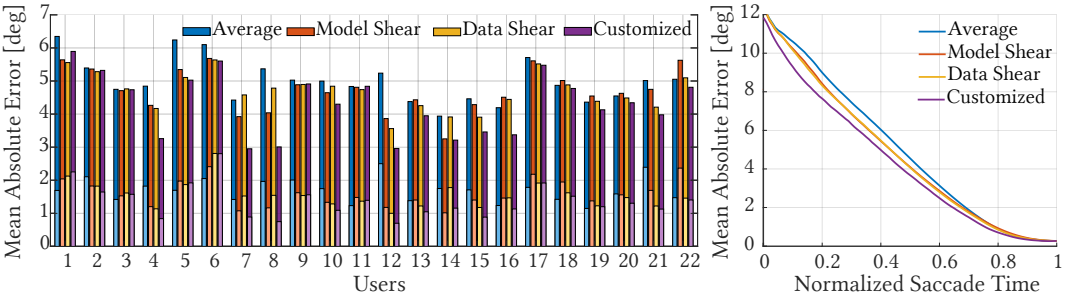


Figure 5.12. The figure presents the performance of different models for each user. The bar plot on the left shows aggregate mean absolute errors for the saccade amplitude predictions. The height of the bars represents the mean error measured for whole duration of the saccade while the segments shaded with lighter colors represent the mean error measured in the second half of the saccade duration. The line plot on the right shows the mean absolute error as a function of point in time when the prediction was made during the saccade. The customized model gives the best performance, followed by model shear and then the average model (please see the text for details). Model shear mostly has a good prediction performance for the users, for whom the customized model also performs well.

Based on the above experiments, we conclude that both the *data* and *model*

shear are viable solutions for extending and improving saccade prediction models to account for effects analyzed in Section 5.4. The important difference between them lies in how they can correct model biases. While the model shear is capable of correcting them, adjusting the data using *data shear* is not. Therefore, the success of the *data shear* is influenced by the quality of the prediction model built upon it.

5.6 Conclusion

In many applications, such as foveated rendering, the latency poses significant challenges. Improving hardware solutions is one path for improving the performance of the techniques that benefit accurate gaze information. However, it has been demonstrated that latency problems can also be addressed by building efficient and accurate predictive models for fast eye movements (Chapter 4). In this chapter, we extend our model and analyze factors that should be accounted for when building such methods. We first demonstrate that factors, which were previously not considered explicitly, such as the orientation of the saccade, depth change, or initial smooth pursuit eye motion, affect the saccade profiles. Then, we propose a technique that allows extending previous models, such as ours, and datasets to train them to handle the additional effects while limiting the number of collected data in user experiments. We argue that this is critical for building comprehensive models for saccade prediction. The key to our technique is the proposed shearing operation which adapts previously derived models. This low parameter transformation acts as a regularization for smaller, possibly more noisy datasets. In this work, we demonstrated the performance of the method on training personalized models and models for horizontal and vertical saccades. In the future, the method can be used to train more comprehensive models addressing a continuous range of orientation, depth changes, user-specific factors, and possibly other factors using a lower number of input saccades. We also believe that the low number of parameters of the shear transformation will allow creating models that will adapt on the fly to the user without the additional need for calibration. Finally, our method can be seen as a data augmentation technique for machine learning techniques, such as [Morales et al., 2018]. While the current inference times do not meet the low latency demand of the state-of-the-art head-mounted displays, such techniques can provide acceptable performance and higher accuracy prediction in the future. In this context, our method can significantly limit the amount of data required for training such models facilitating the development and application of these techniques.

Chapter 6

Luminance-Contrast-Aware Foveated Rendering

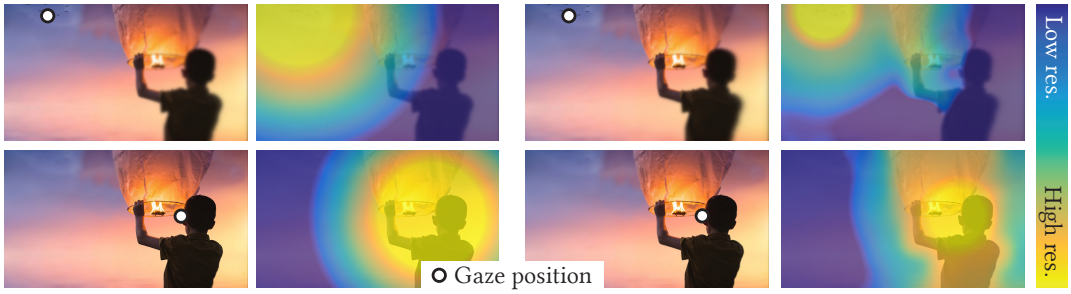


Figure 6.1. Current foveated rendering techniques (left) use a fixed quality decay for peripheral vision. While this can be a conservative solution, it does not provide a full computational benefit. Our technique (right) performs content-adaptive foveation and relaxes the quality requirements for content for which the sensitivity of the human visual system at large eccentricities degrades faster.

6.1 Introduction

In the previous chapters we addressed the problem of system latency, present in the gaze-contingent systems. In this chapter we will explore other options for improving the user experience and we will focus on foveated rendering in particular. Due to the rapid development of low-cost eye trackers, foveated rendering will play a key role for new VR devices [Durbin, 2017]. Although the benefits

of such an approach have been successfully demonstrated for many quality attributes, e.g., spatial resolution, color, and depth (Chapter 3), we show that these techniques do not fully exploit their potential. In particular, most of the existing techniques propose to degrade the rendering quality as a function of eccentricity, but neglect the fact that the sensitivity of the HVS to image distortions also depends on the underlying content - the effect known as visual masking. A relevant observation for our work is that the visibility of foveation depends on the underlying luminance contrast, i.e., while a given reduction of spatial resolution becomes objectionable in high-contrast regions, it remains unnoticed for low-contrast regions (Figure 6.2). As we later show in this chapter (Section 6.4), this observation is confirmed by our measurements for different visual eccentricities, which show a significant difference in the tolerable amount of quality degradation depending on the underlying visual content (Figure 6.7).



Figure 6.2. The same foveation exhibits different visibility depending on the underlying texture. In this image, the foveation was optimized such that it is invisible for the photograph (left part). At the same time, however, it can be easily detected on the text texture (right part).

In this chapter, we exploit the above observation and propose a luminance-contrast-aware foveated rendering strategy. In contrast to previous techniques, our method adjusts the spatial resolution not only according to the eccentricity but also to the underlying luminance information, taking into account the strength of local visual masking. To this end, we propose a new, low-cost predictor that takes a current frame as an input and provides a spatially-varying map

of required spatial resolution. The predictor is based on existing models of visual masking, but it is trained for foveated rendering on a new dataset acquired in a psychophysical experiment. We demonstrate that such prediction can be accurate even if the input is a low-resolution frame. This property is critical, as it allows us to predict the required parameters of foveated rendering based on a crude approximation of the new frame. To apply the prediction in foveated rendering, we first render a low-resolution version of a frame to which we apply our predictor. Next, we render the frame according to the predicted quality. We demonstrate that this strategy leads to substantial computational savings without reducing visual quality. The results are validated in a series of user experiments including simulated foveated rendering system. The main contributions in this chapter include:

- an efficient data collection procedure for testing visibility of foveation for a wide field-of-view,
- perceptual experiments investigating the visibility of spatial resolution reduction as a function of eccentricity and underlying luminance signal for complex image patches,
- an efficient prediction of required spatial resolution based on a low-resolu–tion input frame,
- application of the predictor to foveated rendering in desktop system with eye tracking.

6.2 Overview

Our approach relies on a new computational model for luminance contrast (Section 6.3), which estimates the maximum spatial resolution loss that can be introduced to an image without visible artifacts. It is based on underlying content and eccentricity which are important in the context of foveated rendering. The model relies on characteristics of the HVS such as the peripheral contrast sensitivity and a transducer model for contrast perception. We calibrate the model prediction using our new experimental data (Section 6.4). Our technique relies on two critical observations described below.

Hoffman et al. [2018] and Albert et al. [2017] demonstrate that temporarily-stable low-resolution rendering is perceptually equivalent to a Gaussian-blurred high-resolution rendering. This motivates our technique to model the resolution

reduction using a Gaussian low-pass filter. Consequently, our model uses a standard deviation (σ_s) of the Gaussian filter to express the maximum acceptable resolution reduction. The σ_s value can be later translated into the rendering resolution for given content and used to drive rendering resolution adaptively during real-time rendering (Figure 6.3). Thanks to the above assumption, we derive our model as a closed-form expression, which enables an efficient implementation.

The decision about the optimal rendering resolution would be made best based on full information about the content, i.e., complete contrast information across different spatial frequencies. However, this would require a full-resolution rendering in the first place, and therefore, it is not a feasible solution in foveated rendering. Due to this paradoxical nature of the problem, we first design and test our predictor using high-resolution inputs. Later, we show that it is possible to re-train the model such that it provides the prediction based on a low-resolution rendering. In the latter case, undersampled high-frequency features are still present in a form of aliasing which conveys to our metric information on local contrast localization.

6.3 Computational Model

In this section, we derive a computational model that estimates the maximum resolution reduction that remains undetectable by an observer. The derivation operates on local patches of high-resolution image and computes a standard deviation of a Gaussian low-pass filter which models the resolution degradation. We derive the model in two steps. First, we express the luminance contrast of the patch in perceptual units (Section 6.3.1). Based on this measure, we derive a formula for computing the standard deviation σ_s (Section 6.3.2).

6.3.1 Perceptual Contrast Measure

We express the perceived contrast of a single image patch as a function of spatial frequency and eccentricity. The function accounts for contrast sensitivity of the human visual system as well as visual masking (Figure 6.4). The model, as described here, contains several free parameters which we optimize based on experimental data (Section 6.4).

Luminance contrast The process starts with the conversion of the intensity of every pixel p to an absolute luminance value $L(p)$. Next, we compute band-

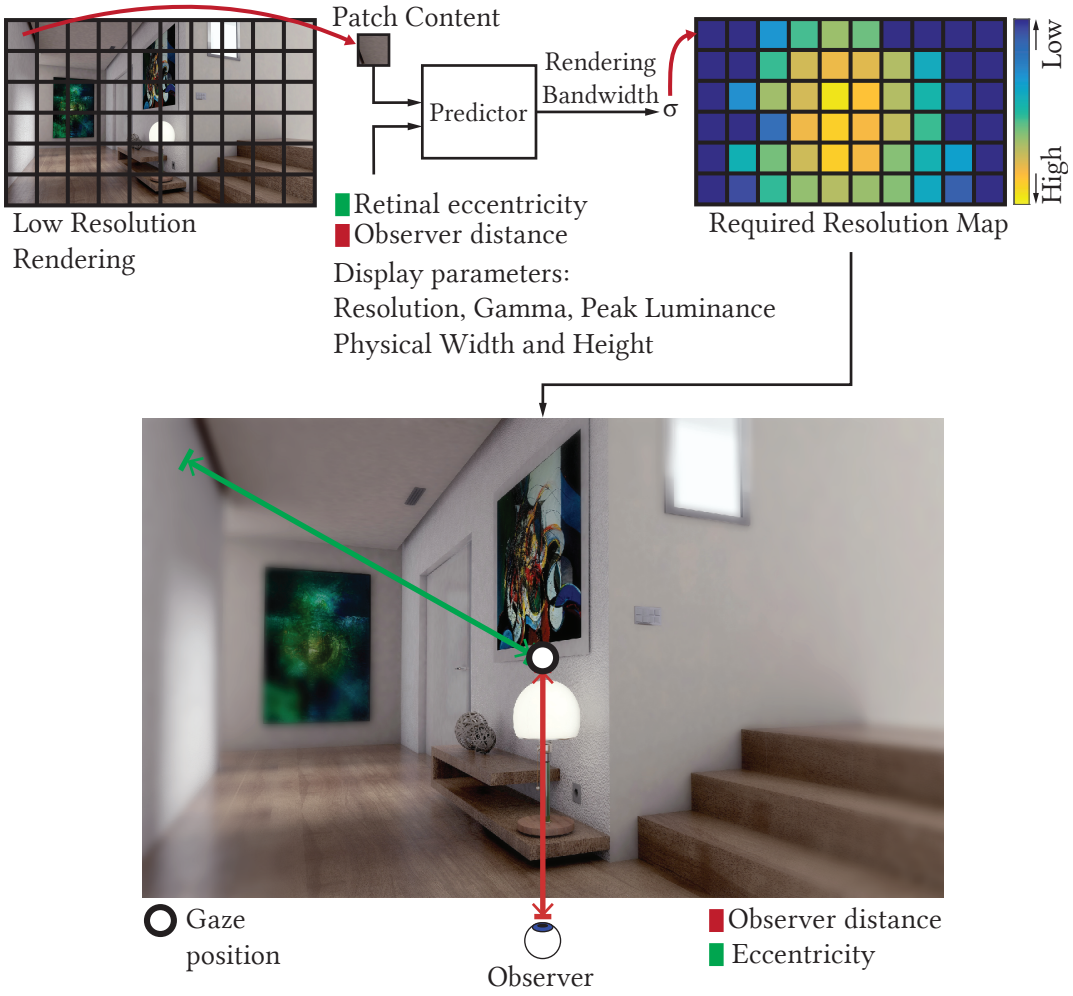


Figure 6.3. Overview of our method. Our predictor takes patches, retinal eccentricity, observer distance and display parameters such as the resolution, gamma, peak luminance, physical width and height as inputs and predicts the required spatial rendering bandwidth expressed as the standard deviation of a low-pass Gaussian filter. The map is generated using our method and the output is enhanced for visibility. Image by Pxhere.

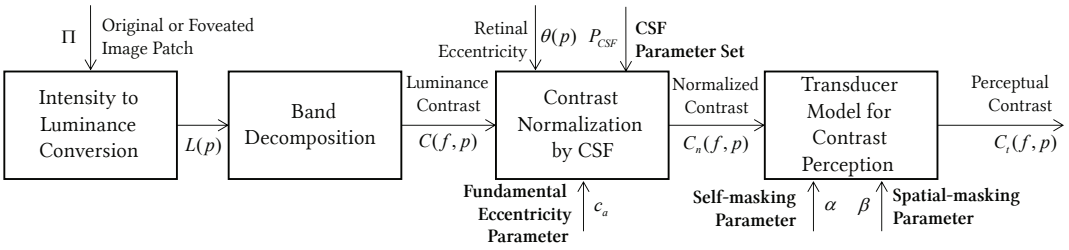


Figure 6.4. This figure shows a flowchart of our model for computing the perceptual contrast measure. The input parameters which are optimized during the calibration are shown in bold.

limited contrast similarly to [Lubin, 1995; Ramasubramanian et al., 1999]. To this end, we first perform a Laplacian pyramid decomposition [Burt and Adelson, 1983] which provides band-limited luminance difference $\Delta L(f,p)$. Then, following [Peli, 1990], we use the decomposition to compute the luminance contrast pyramid as:

$$C(f,p) = \frac{\Delta L(f,p)}{L_a(f,p)+\epsilon}, \tag{6.1}$$

where f is the spatial frequency in cpd units (cycles-per-visual-degree) and ϵ is a small number to prevent mathematical singularities in the regions with low luminance. The average luminance $L_a(f,p)$ in the denominator is provided by the corresponding point in the Gaussian pyramid two levels down in resolution, which is upsampled by a factor of four using a linear interpolation.

Contrast sensitivity and retinal eccentricity To obtain information about the magnitude of perceived contrast, we normalize the values in the pyramid using eccen–tricity-dependent contrast sensitivity function (CSF). This gives us luminance contrast C_n expressed as a multiple of detection threshold:

$$C_n(f,p) = C(f,p)\, S_{\text{CSF}}\big(f, \theta(p), L_a(f,p)\big), \tag{6.2}$$

where $\theta(p)$ is the retinal eccentricity of pixel p expressed in visual degrees, and $L_a(f,p)$ models the adaptation luminance. Here, we base the contrast sensitivity function S_{CSF} on [Peli et al., 1991] where the standard contrast sensitivity function S'_{CSF} for the fovea is attenuated according to the eccentricity:

$$S_{\text{CSF}}(f,\theta,L_a) = \frac{1}{\exp\big(c_a \theta f\big)}\, a(L_a) S'_{\text{CSF}}(f). \tag{6.3}$$

In the above equation, c_a is the fundamental eccentricity parameter that models the rate of HVS acuity loss in the peripheral visual field, f is the spatial frequency,

and $a(L_a) = (1 + 0.7/L_a)^{-0.2}$ represents the effect of adaptation luminance L_a on the peak sensitivity [Barten, 1989]. After initial attempts of using existing CSF definitions such as [Barten, 1989; Mannos and Sakrison, 1974] for S'_{CSF}, we opted for a custom solution. We define the CSF at four frequency bands centered at 4, 8, 16 and 32 cpds with values denoted by s_4, s_8, s_{16} and s_{32} (parameters of our model). The sensitivities for the intermediate frequencies are obtained using cubic Hermite spline interpolation in the log-sensitivity and log-frequency domain. We found that this solution provides a more accurate prediction of our model than using standard CSF functions. We attribute this behavior to a broad-band characteristic of a Laplacian pyramid[1], which is better handled by a custom definition which accounts for broad-band stimuli in contrast to standard CSF which is derived for a single luminance frequency stimuli.

Visual masking In the final step of measuring the perceived luminance contrast, we incorporate the effect of the visual masking. To this end, we use the transducer model of Zeng et al. [2000] on the normalized contrast C_n, and expressed the final value of perceived luminance contrast as:

$$C_t(f,p) = \frac{\text{sign}\big(C_n(f,p)\big) \cdot \big|C_n(f,p)\big|^{\alpha}}{1 + \frac{1}{|N|} \sum_{q \in N(p)} \big|C_n(f,q)\big|^{\beta}}, \tag{6.4}$$

Here, the numerator models the self-masking effects, while the denominator models the spatial-masking effects from the 5×5 neighborhood $N(p)$ in the same band. $\alpha, \beta \in [0,1]$ are parameters of our model, which control the masking modeling.

6.3.2 Estimation of Resolution Reduction

Our goal is to estimate per-patch maximal resolution reduction that would remain unnoticeable by an observer. Since we model the resolution reduction using a Gaussian low-pass filter, we are seeking a maximum standard deviation for the Gaussian filter such that the difference between the original patch and its filtered version will be imperceptible. Using our perceived contrast definition from the previous section (Equation 6.4), we can formalize this problem as:

$$\begin{aligned} &\text{maximize} \quad \sigma_s, \\ &\text{subject to} \quad \forall_{p \in \Pi, f} \; C_t(p,f) - C'_t(p,f) \leq 1, \end{aligned} \tag{6.5}$$

[1] Each band of a Laplacian pyramid contains a broad frequency spectrum, and in particular, the highest frequency band contains a significant portion of medium frequencies.

where $C_t(p,f)$ is the contrast of the original patch Π. In this and the following equations, we use C' notation for all the contrast measures related to the patch Π convolved with G_{σ_s}, a Gaussian function with standard deviation equal to σ_s. Consequently, $C'_t(p,f)$ is the perceived contrast measure of the original patch which is pre-filtered using G_{σ_s}. The constraint in this formulation guarantees that the difference between the two patches will be below the visibility threshold. Due to the complex nature of the contrast definition and the spatial dependencies between contrast values for neighboring regions, the above optimization does not have a direct solution and requires an iterative optimization for the entire image. This would be prohibitively expensive in the context of foveated rendering. Therefore, in this section, we demonstrate how this formulation can be simplified leading to a closed-form solution for σ_s.

Let us first consider estimating σ_s for one pixel p and single spatial frequency f. If the patch Π is convolved with G_{σ_s}, the values in the Laplacian frequency decomposition will be attenuated according to the frequency response of the filter. More precisely, the frequency response of G_{σ_s} for frequency f will be given by:

$$\hat{G}_{\sigma_s}(f) = \frac{C'(f,p)}{C(f,p)}. \tag{6.6}$$

On the other hand, we know that the frequency response of a Gaussian filter G_{σ_s} is also a Gaussian:

$$\hat{G}_{\sigma_s}(f) = \exp\left(-f^2 / \left(2\sigma_f^2\right)\right), \tag{6.7}$$

where σ_f is the standard deviation in the frequency domain, and it is defined as $\sigma_f = (2\pi\sigma_s)^{-1}$. By combining Equations 6.6 and 6.7, one can show that σ_f can be expressed as:

$$\sigma_f = \frac{f}{\sqrt{-2\ln\left(\frac{C'(f,p)}{C(f,p)}\right)}} = \frac{f}{\sqrt{-2\ln\left(\frac{C'_n(f,p)}{C_n(f,p)}\right)}}, \tag{6.8}$$

where the last transition is a direct consequence of the Equation 6.2. In the above equation, $C_n(f,p)$ can directly be computed from the input patch. So the only unknown, besides σ_f which we need to compute, is $C'_n(f,p)$. To obtain its value, we will use the contrast loss constraint from Equation 6.5.

We can assume that when σ_s increases, and so does the difference between $C'_t(f,p)$ and $C_t(f,p)$. Thus, when σ_s is a solution to our problem, the following equality holds: $C'_t(f,p) - C_t(f,p) = 1$. We can directly express perceived contrast

in this equality using Equation 6.4, and obtain:

$$\frac{\operatorname{sign}\big(C_n(f,p)\big)\cdot\big|C_n(f,p)\big|^{\alpha}}{1+\frac{1}{|N|}\sum_{q\in N(p)}\big|C_n(f,q)\big|^{\beta}}-\frac{\operatorname{sign}\big(C'_n(f,p)\big)\cdot\big|C'_n(f,p)\big|^{\alpha}}{1+\frac{1}{|N|}\sum_{q\in N(p)}\big|C'_n(f,q)\big|^{\beta}}=1. \tag{6.9}$$

It becomes clear, that $C'_n(f,p)$ cannot be computed directly from this equation due to the visual spatial masking term in the denominator of the first component. Therefore, we make one more assumption. We assume that the spatial masking for the path convolved with G_{σ_s} can be approximated by the spatial masking in the original patch. Assuming additionally that the sign of the contrast does not change during the filtering, the above equation can be simplified to:

$$\frac{\operatorname{sign}\big(C_n(f,p)\big)\cdot\Big(\big|C_n(f,p)\big|^{\alpha}-\big|C'_n(f,p)\big|^{\alpha}\Big)}{1+\frac{1}{|N|}\sum_{q\in N(p)}\big|C_n(f,q)\big|^{\beta}}=1. \tag{6.10}$$

From the above equation, $C'_n(f,p)$ can directly be derived as:

$$C'_n(f,p)=\left|\big|C_n(f,p)\big|^{\alpha}-\left(1+\frac{1}{|N|}\sum_{q\in N(p)}\big|C_n(f,q)\big|^{\beta}\right)\right|^{1/\alpha}. \tag{6.11}$$

Please note that we omit the sign of the contrast since we are interested only in its magnitude. The above definition of $C'_n(f,p)$ and Equation 6.8 provide a closed-form expression for computing optimal σ_f for a particular pixel p and spatial frequency f.

Now, we could simply use the relation $\sigma_f=(2\pi\sigma_s)^{-1}$ to convert σ_f to the primary domain. Before doing this, we first compute σ_f for entire patch, which is critical for our calibration procedure (Section 6.4). To this end, we first combine σ_f estimation for pixel patch by taking the maximum value across all frequency levels. This allows us to make our method conservative and not overestimate the acceptable resolution reduction. Note that, larger σ_f corresponds to smaller blur, and therefore, smaller acceptable resolution reduction. Next, we combine the obtained values across the entire patch using a smooth maximum function:

$$\hat{\sigma}_f=\left(\sum_{p\in\Pi}\sigma_f(p)\cdot\exp\big(\omega\cdot\sigma_f(p)\big)\right)\Big/\left(\sum_{p\in\Pi}\exp\big(\omega\cdot\sigma_f(p)\big)\right), \tag{6.12}$$

where $\omega\in[0,\infty)$ is a parameter which controls the behavior of the function ranging from computing average as $\omega\to 0$ and maximum as $\omega\to\infty$. When experimenting with optimizing parameters of our model, we found that the smooth

maximum performs better than simply taking maximum value. Finally, σ_s for the entire patch is computed as:

$$\hat{\sigma}_s = \frac{1}{2\pi\hat{\sigma}_f}. \tag{6.13}$$

6.4 Calibration

Our model is defined using several free parameters: self-masking parameter (α), spatial-masking parameter (β), CSF parameters (s_4, s_8, s_{16}, s_{32}), fundamental eccentricity (c_a), and smooth max parameter (ω). In this section, we present a calibration procedure and perceptual experiments that are used to collect necessary user data.

Training our model requires a set of patch pairs consisting of a high-resolution patch as well as its low-resolution version for which the quality degradation is not detectable. One way of collecting such data is measuring maximum and undetectable resolution reduction for individual patches and eccentricities. However, such procedure limits each trial to a single eccentricity and patch. As a result, it requires long sessions to collect data. Instead, we propose to gather the data in a more efficient way. We tile one patch into an image covering the entire screen and estimate the optimal, unnoticeable foveation. This allows us to derive necessary information for a whole range of eccentricities.

We define foveated rendering using two parameters. The first one is the radius r of the foveal region where the visual content is rendered in the highest resolution. The second parameter is the rate k at which the resolution is reduced towards the periphery. Consequently, the resolution reduction modeled by using a standard deviation of a Gaussian filter can be expressed as:

$$\sigma_s(\theta) = \begin{cases} 0, & \text{if } \theta < r, \\ k \cdot (\theta - r), & \text{if } \theta \geq r, \end{cases} \tag{6.14}$$

where θ is the retinal eccentricity of a particular point on the screen. Different combinations of r and k affect the tradeoff between quality reduction and rendering efficiency. The goal of our experiment is to measure the visibility of foveation for different parameters and image patches to determine the strongest one which remains unnoticeable.

The experimental setup Our system consists of a Tobii TX300 Eye Tracker, a chin-rest to keep the observation point relatively stable during the experiments, and

two displays. The first one is a 27" ASUS PG278Q display with 2560 × 1440 resolution spanning a visual field of 48.3° × 28.3° from a viewing distance of 66.5 cm. The second display is a 32" Dell UP3216Q with 3840 × 2160 resolution spanning a visual field of 52.3° × 30.9° from a viewing distance of 71 cm. The peak luminances of the displays are measured as 214.6 cd/m^2 and 199.2 cd/m^2 whereas the peak resolutions produced at the center are 24.9 cpd and 34.1 cpd for the first and the second displays, respectively.

The stimuli Our dataset consists of 36 patches selected from natural and synthetic images with different characteristics (see Figure 6.5). We picked the first 18 patches randomly from a large set of 5640 natural images [Cimpoi et al., 2014]. To improve the diversity of our subset, the remaining 18 patches were picked from the large dataset by maximizing the dissimilarity between patches by choosing every patch in a greedy fashion using the following formula:

$$I_{n+1} = \arg\max_{I} \{ d(I, d_n) \}, \tag{6.15}$$

where d_n is the dataset consisting of the first n patches and I_{n+1} is the next patch that is added to the dataset. The dissimilarity measure $d(I, d_n)$ between a candidate image and the dataset is defined as

$$d(I, d_n) = \sum_{k=1}^{K} \left| l_k(I) - \frac{1}{|d_n|} \sum_{I_d \in d_n} l_k(I_d) \right| \tag{6.16}$$

where $l_k(I)$ is the mean absolute deviation of pixels in kth level of Laplacian pyramid for image I. This dissimilarity metric maximizes the diversity of frequency content in the dataset by picking the image which has the Laplacian decomposition least similar to the average of existing images.

For the final stimuli, we fill the display frame by tiling an input patch. We avoid introducing high-frequency components on the transitions between tiles by mirroring them about the vertical axis, when tiling in the horizontal direction, and about the horizontal axis, when tiling in the vertical direction (see Figure 6.6). The foveated version of the stimuli are prepared by filtering using a Gaussian kernel with standard deviation given by Equation 6.14. We used 9 different combinations of r and k ($r \in \{4, 7, 11\}$ and $k \in \{0.0017, 0.0035, 0.0052\}$).

The procedure We use a 2AFC procedure, where the alternatives are foveated and non-foveated versions of the same stimuli. Participants were asked to choose

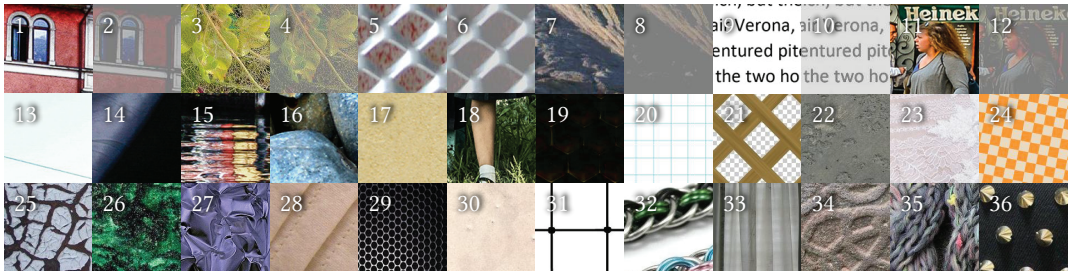


Figure 6.5. Our dataset for the calibration of our predictor. We include patches with different luminance and texture patterns from a dataset of natural and synthetic images [Cimpoi et al., 2014]. Note that patches 1-12 contain reduced-contrast versions of the same content to cover a wider range of contrasts during the calibration phase.

the image which did not contain foveation. In order to get the participants familiar with the experiment interface and controls, we included a training stage at the beginning of the experiment, where the concept of foveation was explained using an exaggerated example. The stimuli were prepared with the assumption that the gaze position is located at the center of the display. The stimulus was hidden when the gaze position deviated from the center. Different alternatives were indicated by randomly assigned letters (A and B) at the center of the screen. The observers were able to switch between two alternatives freely, and they were shown a uniform gray screen in-between. A total of 8 participants with normal or corrected-to-normal vision participated in our experiment, and they made a total of 324 comparisons in six sessions for 36 patches. Each comparison was repeated 10 times by each participant and the average time required for a participant to complete a single session was 40 minutes.

Results From the results of the above experiment, we want for each patch to compute an optimal σ_s as a function of eccentricity. To this end, we first compute for each patch i the probability of detecting foveation given by triplet (r, k, θ) as:

$$P(det|r,k,\theta) = \frac{1}{N}\sum_{n=1}^{N} a_n(r,k,\theta), \tag{6.17}$$

$$a_n(r,k,\theta) = \begin{cases} 1, \text{if non-foveated stimulus is chosen,} \\ 0, \text{otherwise.} \end{cases} \tag{6.18}$$

where N is the number of comparisons by each participant. If $P(det|r,k,\theta) < 0.75$, we labeled this combination of (r, k, θ) as undetectable. We then for each

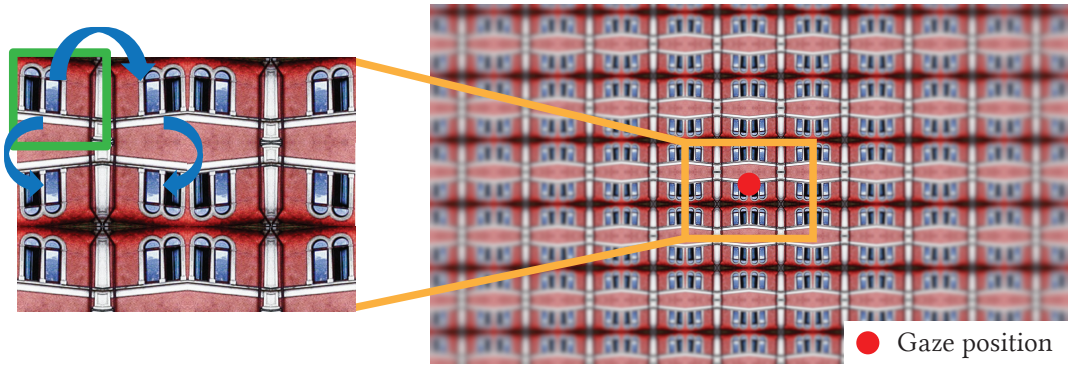


Figure 6.6. A sample stimulus used for data collection and calibration. The zoomed region shows how the input patch is tiled prior to Gaussian filtering. For different values of foveal region radius r and rate of quality drop-off k, the participants are asked to compare foveated (shown here) and non-foveated (without Gaussian blur) versions in a 2AFC experiment.

eccentricity take the maximum value of σ_s across all (r, k, θ) marked as undetected. This defines per-patch and per-observer optimal $\sigma_s(\theta)$. As the last step, we average $\sigma_s(\theta)$ values across the participants to obtain the ground truth $\sigma_s^{(i)}(\theta)$ for patch i. The same procedure is repeated for all patches. The resulting $\sigma_s^{(i)}$ functions are shown in Figure 6.7. The range marked by the whiskers indicates to what extent the acceptable blur depends on the underlying patch for a particular eccentricity. The significant differences between sigma values for different patches (see insets) are the central insight we use in our work. In Figure 6.8 and Figure 6.9, we show the effect of content on the detection threshold for foveated quality degradation. These plots provide the change in the mean and standard deviation of $\sigma_s^{(i)}$ for each patch as a function of eccentricity from the data collected in our subjective experiment for model calibration.

In our implementation, we choose detection threshold, 0.75, as the middle value between the success rate associated with random guessing ($P(det|r, k, \theta) = 0.50$ with two alternatives) and the probability of a guaranteed detection. which is $P(det|r, k, \theta) = 1.00$. This value is commonly used in previous perceptual studies to estimate "barely" visible differences [Lubin, 1995]. The threshold can be adjusted based on the application requirements, and the ground truth associated with a different threshold probability can be easily computed from existing data without repeating the perceptual experiment.

Optimization Finally, to find the parameters of our model, we use a hybrid optimization approach. In the first stage, we use Adaptive Simulated Annealing

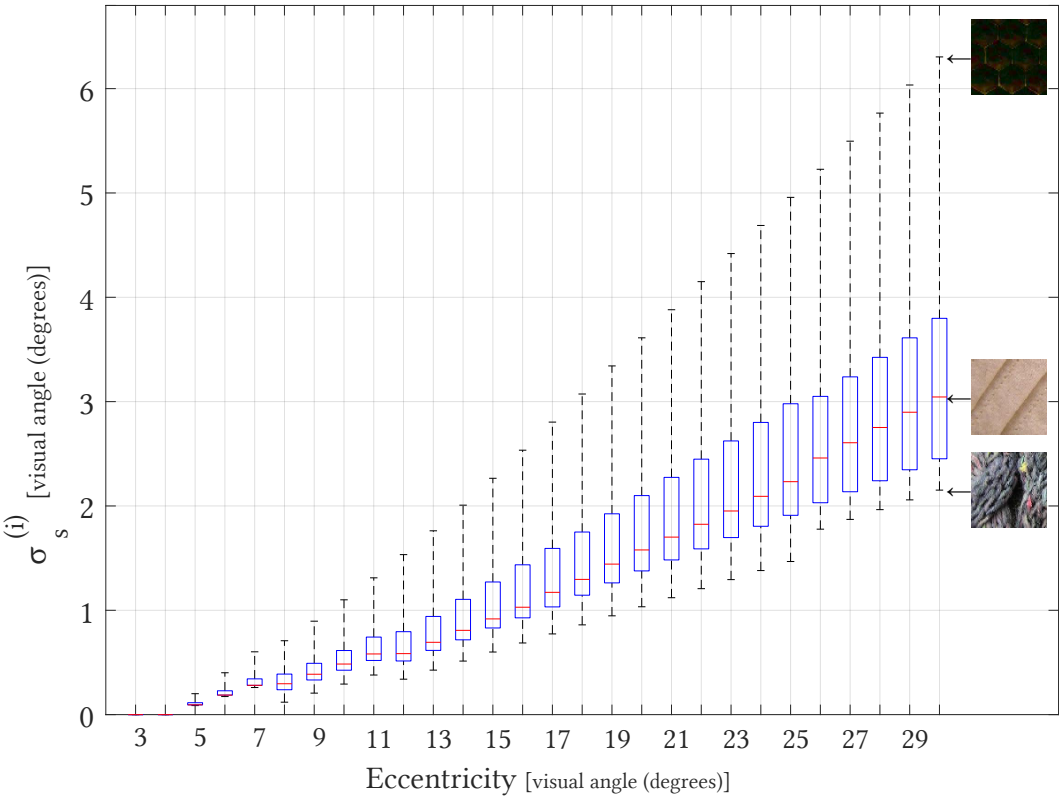


Figure 6.7. Box plot of ground truth $\sigma_s^{(i)}$ obtained from our experiment. This plot shows how content influences the tolerable amount of foveation with respect to eccentricity. Red lines represent the median while boxes show the range between 25th and 75th percentile of the data. Whiskers extend to the whole range. The patches which have the minimum, the median and the maximum $\sigma_s^{(i)}$ are shown on the plot for 30° eccentricity.

(ASA) [Aguiar e Oliveira Junior et al., 2012] to optimize for predictor parameters. In the second stage, we run a gradient-based minimization to fine-tune the result of ASA. This hybrid optimization scheme helps avoiding local optima. The following weighted Mean Absolute Error (MAE) is minimized during the optimization:

$$E = \min_{\mathbb{S}} \frac{1}{36} \sum_{i=1}^{36} \sum_{\theta=4^\circ}^{30^\circ} w_1(\theta) \left| w_2\left(\hat{\sigma}_s^{(i)}(\theta) - \sigma_s^{(i)}(\theta)\right)\right|, \tag{6.19}$$

where $\mathbb{S} = \{\alpha, \beta, c_a, s_4, s_8, s_{16}, s_{32}, \omega\}$ is the set of model parameters, $\sigma_s^{(i)}(\theta)$ is the ground truth for patch i from our experiment and $\hat{\sigma}_s^{(i)}(\theta)$ is the result of our

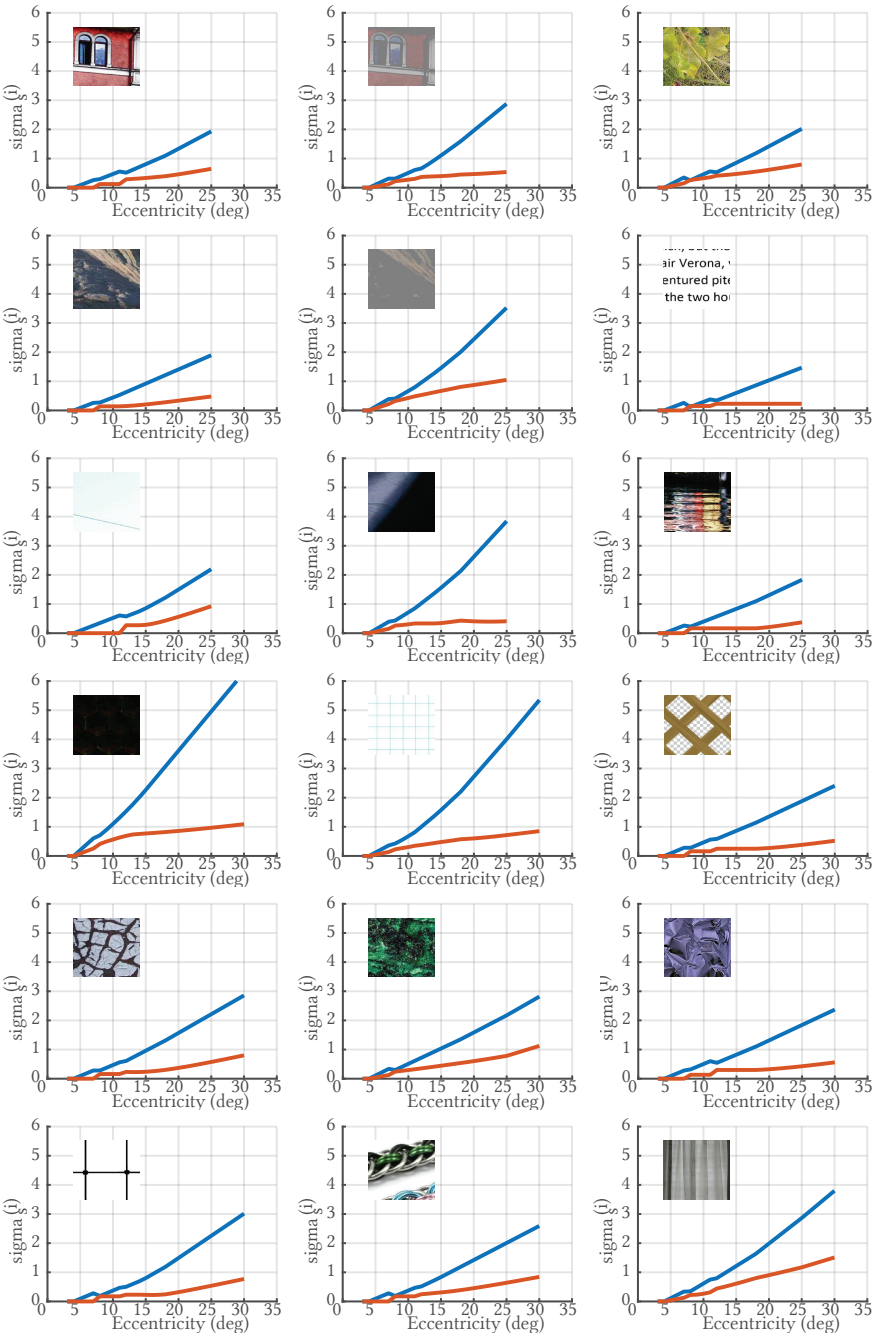


Figure 6.8. These plots show how $\sigma_s^{(i)}$ (y-axis) changes with respect to eccentricity (x-axis) for each patch. The lines represent the mean (blue) and the standard deviation (red) for the given patch across all participants.

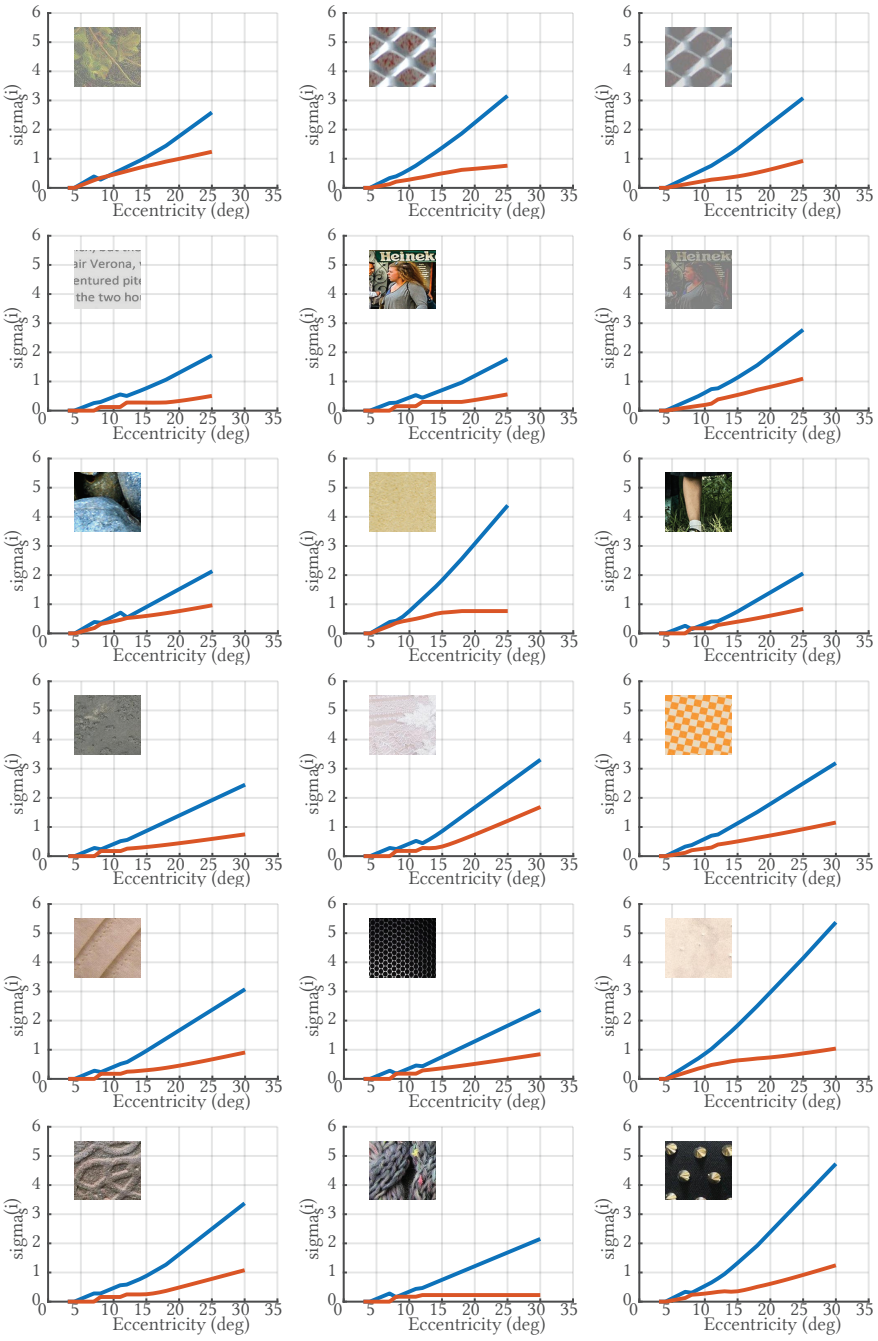


Figure 6.9. These plots show how $\sigma_s^{(i)}$ (y-axis) changes with respect to eccentricity (x-axis) for each patch. The lines represent the mean (blue) and the standard deviation (red) for the given patch across all participants.

predictor for the eccentricity θ. w_1 and w_2 are weighting functions defined as:

$$w_1(\theta) = \begin{cases} 2, & \text{if } \theta < 10, \\ 1, & \text{otherwise} \end{cases} \tag{6.20}$$

$$w_2(x) = \begin{cases} 8x, & \text{if } x > 0, \\ x, & \text{otherwise.} \end{cases} \tag{6.21}$$

The first function, w_1 puts more emphasis on the error measured in the parafoveal region where HVS has a higher sensitivity. On the other hand, the second function, w_2, penalizes underestimation of spatial bandwidth with a larger weight, because underestimation is less desirable than overestimation due to potential visual artifacts.

Table 6.1. Optimal parameter values obtained during calibration and corresponding cross-validation errors.

		Parameters								Loss Func.		MAE	
	CV	α	β	c_a	s_4	s_8	s_{16}	s_{32}	ω	Train	Test	Train	Test
Full Resolution	1	0.46	0.16	0.04	5.62	6.00	7.48	6.06	1.89	0.62	0.79	0.50	0.26
	2	0.57	0.13	0.04	4.50	6.01	3.60	4.00	1.81	0.73	0.86	0.65	0.63
	3	0.53	0.28	0.04	6.31	5.90	7.82	6.12	1.76	0.62	0.56	0.51	0.53
	4	0.48	0.23	0.02	4.66	6.51	2.12	8.00	1.48	0.65	0.60	0.56	0.57
	5	0.54	0.29	0.04	6.29	5.93	7.81	5.96	1.82	0.58	0.75	0.48	0.70
	6	0.49	0.24	0.04	6.47	6.04	7.99	6.25	1.87	0.59	0.66	0.50	0.60
	All	0.61	0.39	0.04	6.30	5.55	7.48	7.95	1.59	0.61	-	0.51	-
Downscaled	1	0.57	0.11	0.04	4.94	5.89	3.81	4.00	1.16	0.74	0.88	0.60	0.33
	2	0.55	0.12	0.04	4.85	6.09	2.99	4.00	1.67	0.69	0.94	0.59	0.61
	3	0.55	0.13	0.04	5.37	6.23	4.07	4.07	1.63	0.72	0.65	0.62	0.61
	4	0.56	0.13	0.04	5.27	6.17	3.18	3.31	1.53	0.70	0.74	0.61	0.72
	5	0.55	0.13	0.04	5.38	6.24	4.11	4.07	1.62	0.68	0.84	0.58	0.75
	6	0.52	0.11	0.04	5.38	6.43	3.27	4.87	1.64	0.69	0.80	0.58	0.72
	All	0.55	0.14	0.04	5.29	6.23	3.40	4.01	1.55	0.71	-	0.55	-

We check the generalized performance by performing 6-fold cross-validation. Optimal parameters and errors measured at each fold are depicted in Table 6.1, where we provide a detailed list of parameter values obtained from the optimization. We observe that the test errors are close to training errors and optimal

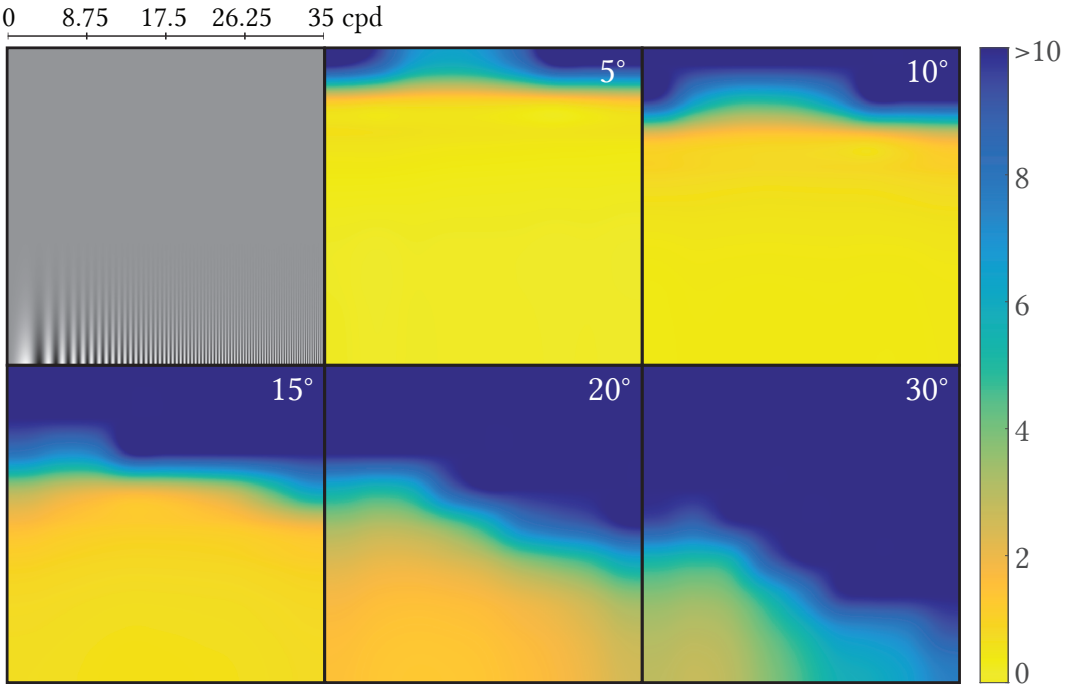


Figure 6.10. We used a Campbell-Robson chart (top left) as a test input for our predictor. The predictions of σ_s from our model are given for different visual eccentricities. The eccentricities are indicated at the top-right corner of each map. Our model successfully predicts a higher σ_s (corresponding to a lower rendering resolution) for the spatial frequencies that are imperceptible by the HVS as the visual eccentricity increases and the contrast declines. (Please note that the Campbell-Robson chart is prone to aliasing when viewed or printed in low resolution. Please refer to the electronic copy of this document for a correct illustration.)

parameter values are stable among different cross validation folds. As expected, higher training and testing errors are observed when the predictor is calibrated on the inputs with reduced resolution due to the loss of information on high-frequency bands. But we can still assume a reasonable approximation by the predictor due to the small difference in MAE (0.503 compared to 0.554). The optimal parameter values that we are using in our validation are obtained by calibrating our predictor using the whole dataset. These values are given in Table 6.2.

As explained in Section 6.2, a method for predicting the acceptable resolution degradation operating on a full-resolution image is not very useful in the context of foveated rendering. Therefore, we take advantage of our parametric

design of the model and train it on low-resolution images, which can be provided by foveated rendering as a cheap initial approximation of the frame. Consequently, we calibrate and run our model on inputs which are downsampled by a factor of 1/4 (corresponding to 1/16 of the target area). The downsampling routine employed during the calibration is nearest-neighbor downsampling and it does not involve low-pass filtering or interpolation. This is equivalent to actual low-resolution rendering, including spatial aliasing effects that may arise during rendering. This way, our model is able to utilize cues which appear in the form of aliasing in the presence of higher-frequency content. When actual rendering takes place, we render true low resolution. This gives the optimal performance for the actual rendering applications.

Table 6.2. Best parameter values obtained after calibration. The input patches are downscaled by a factor of 1/4. Loss is the training error computed using Equation 6.19. In addition to the loss function, which is a weighted mean absolute error, we also provide the standard unweighted mean absolute error (MAE) for evaluation.

α	β	c_a	$\log_{10}(s_4)$	$\log_{10}(s_8)$
0.555	0.135	0.040	5.290	6.226
$\log_{10}(s_{16})$	$\log_{10}(s_{32})$	ω	Loss	MAE
3.404	4.011	1.919	0.706	0.554

6.5 Implementation

For our validation experiments, we implemented our method in C++ using the OpenGL Shading Language (GLSL). We adapted our implementation of the model described in Section 6.3 to fully benefit from the optimizations in GLSL. For example, we implement local operations defined on patches and pixels (such as those given in Equations 6.1-6.13) in a way that allows the graphics card to process the whole frame in parallel. Similarly, the decomposition into different frequency bands is achieved using mipmaps to maximize the parallelism, where higher levels represent lower frequencies. σ_f estimation from different frequencies are combined by performing a level-of-detail texture lookup in the pyramid for efficiency. On the other hand, the smooth max function is executed by computing the mipmap of the optimal standard deviation and taking the level in the pyramid which corresponds to the maximum achievable level for a single patch. In all operations, we preserve the patch-pixel correspondence to maintain locality

information. We used different patch sizes for two displays during calibration; namely, 128 × 128 for the Asus display (2560 × 1440) and 192 × 192 for the Dell display (3840 × 2160). For rendering, 128 × 128 was used (corresponds to 32 × 32 effective patch size for 1/4 downsampled inputs). The choice of patch size is mainly dependent on the number of bands required in Band Decomposition step. Using smaller patches brings a limitation on the pyramid decomposition while using larger patches makes the predictions less sensitive to local changes in content. A patch size of 128 × 128 provides a good balance between these two.

Performance Table 6.3 shows the performance benchmark of the predictor implementation on an NVIDIA GeForce GTX 2080Ti graphics card. Since our setup consists of two different displays we provide the measurements for their respective resolutions. For comparison, we also include the time measurements using full resolution inputs. In our validation experiments, we run our predictor on inputs which are downscaled by a factor of 1/4 (corresponding to 1/16 of the target area). This gives the optimal performance for the actual rendering applications. For comparison, we also include the time measurements using full resolution inputs.

Table 6.3. Running times of our implementation. Our predictor is calibrated and validated using 1/4× downsampled inputs (1/16× of the area) in a series of subjective experiments. Here, we show the computational savings obtained by our approach with respect to the predictions from full-resolution inputs. Please note that these values do not include rendering costs.

Input Size	2560 × 1440	3840 × 2160
Downsampled (1/4 ×)	0.7 ms	1.2 ms
Full resolution	3.0 ms	5.9 ms

6.6 Validation

In this section, we first show the results from rendering bandwidth predictions from our method for different inputs. Then we present our results from two subjective experiments. In the first experiment, the participants compare our method with non-foveated rendering. In the second one, they compare our lo-

cally adaptive content-aware foveation method with a globally adaptive foveation strategy.

6.6.1 Visual Evaluation

In Figure 6.11, outputs of our predictor for 12 different inputs are shown. Overall, we observe that the model successfully adapts the rendering bandwidth to the content, while taking the peripheral sensitivity loss of the HVS into account. In inputs 1, 2 and 4-6, we mostly see the effect of defocus in images with different color, luminance and texture content. For these inputs, our method suggests a higher rendering resolution (represented by a lower $\hat{\sigma}_s$ prediction) on the objects which are in focus. In images 3, 7 and 12, we observe that a significant amount of rendering budget is allocated to the buildings, which contain a large amount of detail, and a much lower amount of the bandwidth is allocated to the mostly uniform regions in the sky. It is possible to see how the heatmap adapts to the silhouette of the street lamp in image 7 and the baloon in image 10, which appear as objects with high level of detail on a region with low detail. This rendering scheme minimizes the potential losses in perceived visual quality during rendering and allows providing a considerably higher level of perceived quality by efficient allocation of rendering resources.

6.6.2 Foveated vs. Non-Foveated Rendering

The ground truth that we use for calibration corresponds to a perceived contrast loss with a detection probability of 0.75 (1 JND). If the calibration is successful, the contrast loss in the outputs of our method should be detected approximately with this probability. In order to validate this behavior, we perform a 2AFC subjective experiment, where 12 participants are asked to compare the results of foveated rendering based on our method's prediction and non-foveated rendering. We used the images given in Figure 6.11 as the experiment stimuli. We conduct the experiment using a multiplier of 0.5, 1 and 2 on the predictions of our method ($\hat{\sigma}_s$) to distinguish between random guessing behavior and actual detection by participants. Input stimuli are shown in randomized order and the participants are unaware of the multiplier value of each stimuli during the experiment. In actual detection, we expect to see an increasing detection rate with the multiplier because a larger $\hat{\sigma}_s$ would result in over-blurring whereas such an increasing trend in the detection rate would not be observed in random guessing. The results of this experiment are shown in Figure 6.12. We observe an increasing trend in the detection rates as expected. On the other hand, the detection rate

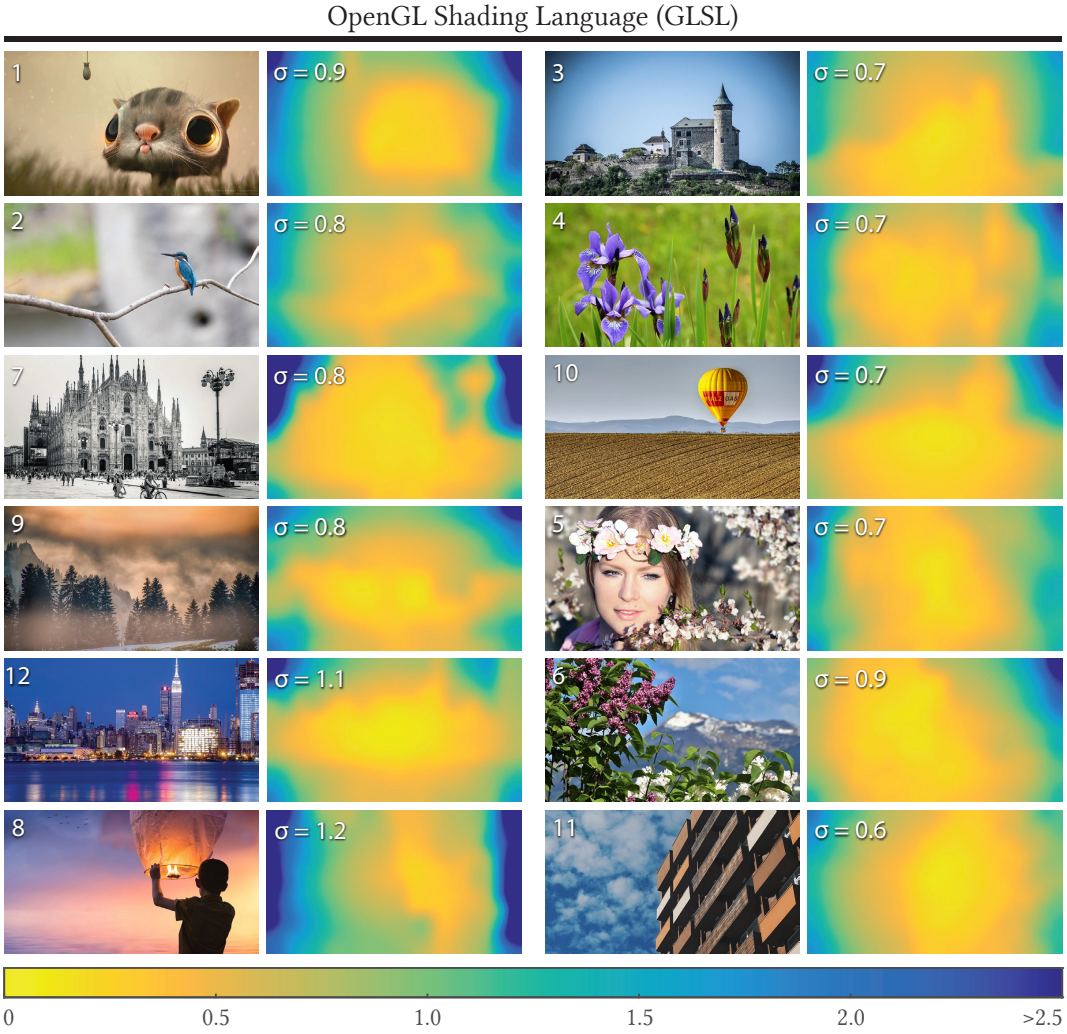


Figure 6.11. The outputs from our model for the images used in our validation experiments. The gaze position is fixed at the center for all images. The heatmaps (right) show the predicted standard deviations ($\hat{\sigma}_s$) of a low-pass Gaussian filter which results in a contrast loss of 1 JND when applied on the input (left). $\hat{\sigma}_s = 0$ represents a requirement for rendering in the native display resolution whereas larger values represent rendering in a lower resolution. Average value of each $\hat{\sigma}_s$ map is shown in the top-left corner. These results show how the effect of content is captured by our method to adapt foveation strength for rendering.

Image 1 is by Manu Jarvinen.

for the predictions $\hat{\sigma}_s$ (multiplier = 1), is lower than 0.75. We attribute having a lower detection rate to our custom loss function during calibration (Section 6.4), which penalizes overestimations with a larger weight.

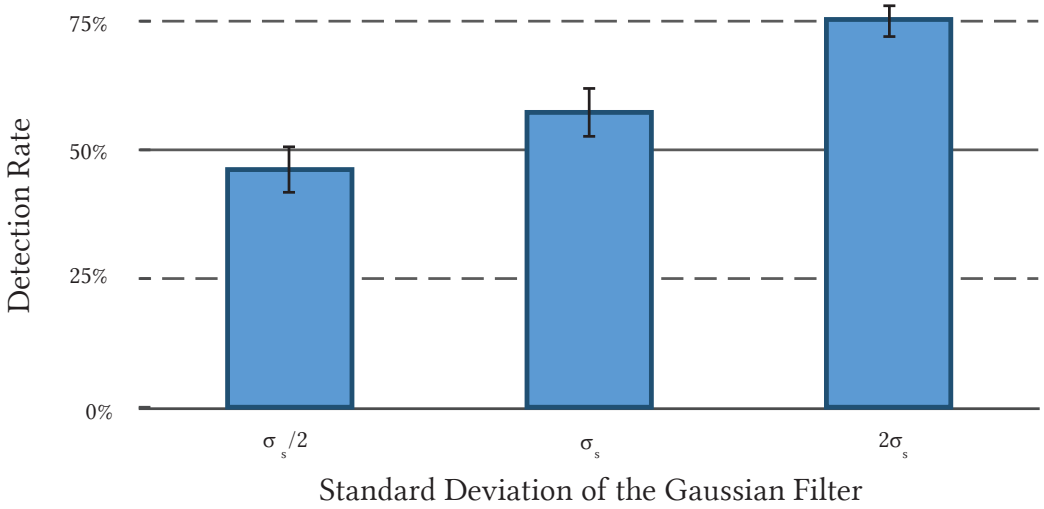


Figure 6.12. Detection rates of the participants for our method and non-foveated rendering. x-axis represents different multipliers that we use for changing the average σ_s prediction to test the effect of different rendering budgets on the preferences of participants. The actual prediction of our method corresponds to the multiplier value of 1 and increasing values on the x-axis represent more limited rendering budgets. The trend in the detection rate shows that the participants actually detect the foveation and the detection rate for the actual rendering is smaller than 0.75 for all platforms when the predictions are not scaled. The error bars represent standard error.

6.6.3 Local vs. Global Adaptation

A suboptimal alternative to our method would be to adjust the foveal region radius in visual degrees, r, and the rate of resolution drop-off in the periphery, k, depending on the content of the whole frame (see Section 6.4 for definitions of r and k). This approach does not take into account local changes in the contrast; therefore, it is a globally adaptive foveated rendering. In a 2AFC experiment, we analyze the visual quality provided by our method (local adaptation) and the globally adaptive foveated rendering. At the beginning of the experiment, we run our method on input images which are provided in Figure 6.11 and compute the average standard deviation of foveation kernel $\bar{\sigma}_s$. Then we ask 12 participants

to choose the optimal foveal region radius, $r \in \{4, 7, 11\}$, that provides the best visual quality in terms of overall sharpness for each image. At the end of this procedure, we compute the rate of resolution drop-off, k, as the value which gives the same average $\bar{\sigma}_s$ to have an equal amount of rendering cost in both methods. The average foveal region sizes from the preferences of our participants are shown in Figure 6.14.

Next, we asked the participants to compare local and global adaptation and choose the stimuli which offers the best overall sharpness. A total of 144 comparisons were made on 12 images given in Figure 6.11. As a result of this experiment, we observe a significant amount of difference in the preferences of participants towards our method. In total, our method is preferred in 100 of 144 comparisons ($p < 0.001$, Binom. test). The preferences of the participants for each image are given in Figure 6.13. For the majority of the images, our method is preferred over the global adaptation with the exception of images 5 and 6. Image 5 is a portait and we believe that the presence of a human face might be playing a special role by increasing the sensitivity of HVS to some distortions. In image 6, the distribution of rendering resolution from our method resembles the result obtained from global adaptation. We think that due to the similar foveation results from two methods, the participants were mostly indifferent between two methods for this image.

6.6.4 Further validation

In our publication Tursun et al. [2019] the method is additionally validated using the Unity3D game engine [Unity3D, 2018] on both desktop and HMD displays and using NVIDIA Variable Rate Shading (VRS) API Nvidia [2018] with OpenGL on desktop display. The results from these experiments are mostly consistent with the results described in this section. As a proof of concept, we also implemented a custom foveated raytracer using OpenCL with RadeonRays routines, which was guided by our method. These additional evaluations are not part of this dissertation.

6.7 Limitations and Future Work

In our work, we use Gaussian blur to model quality degradation due to foveated rendering. While this allowed us to derive a closed-form solution to the problem of finding the optimal foveation, it is only an approximation. We believe, however, that the accuracy of prediction will still hold for small deviations from this

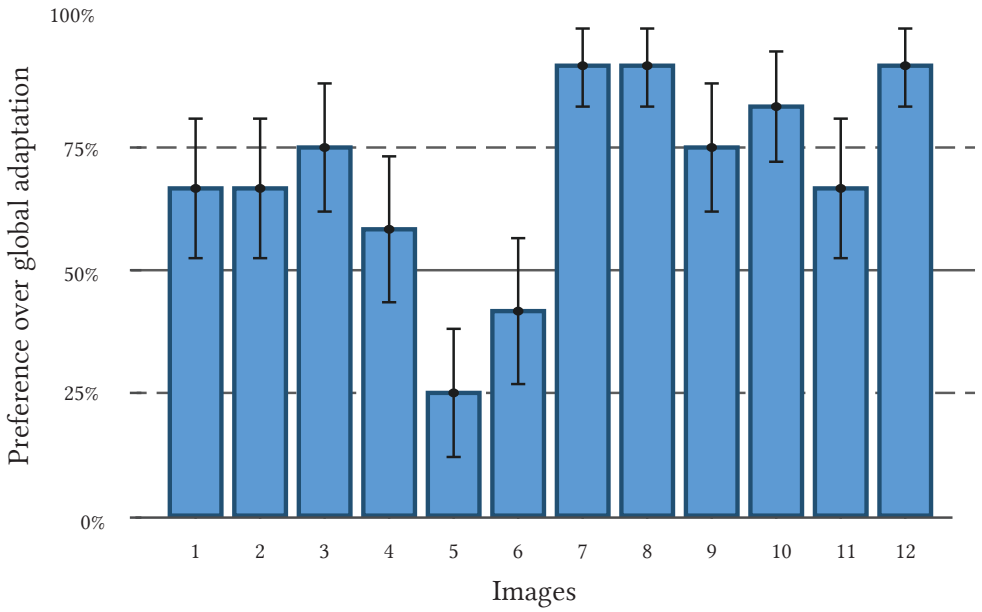


Figure 6.13. The result of our subjective experiments where the participants compared our method with globally adaptive foveated rendering, which does not take local distribution of contrast into account. The error bars represent standard error.

assumption, and when a better accuracy is needed, our method can be retrained in the future following the strategy proposed in this chapter. Another exciting direction for future work is to consider temporal aspects of foveated rendering. It is known that motion also reduces the sensitivity of the human visual system, and including this factor might be beneficial.

Our initial data collection procedure for calibration had a simpler design, where the stimuli consisted of a single patch displayed at a selected eccentricity from a pre-defined set in each trial. The process turned out to be prohibitively time-consuming, and it did not simulate well the case of foveated rendering because each patch was viewed in isolation on a uniform background. To improve the efficiency of data collection and to make the stimuli more realistic, we decided to perform experiments using stimuli filling the entire screen. This procedure allowed us to collect data simultaneously for an extensive range of eccentricity. Tiling the patches may introduce additional luminance frequencies which are not present in the original patch. We minimized this problem by flipping the patches. Furthermore, our experiment was performed for a limited set of (r, k)-pairs. Even though a denser sampling could lead to more accurate

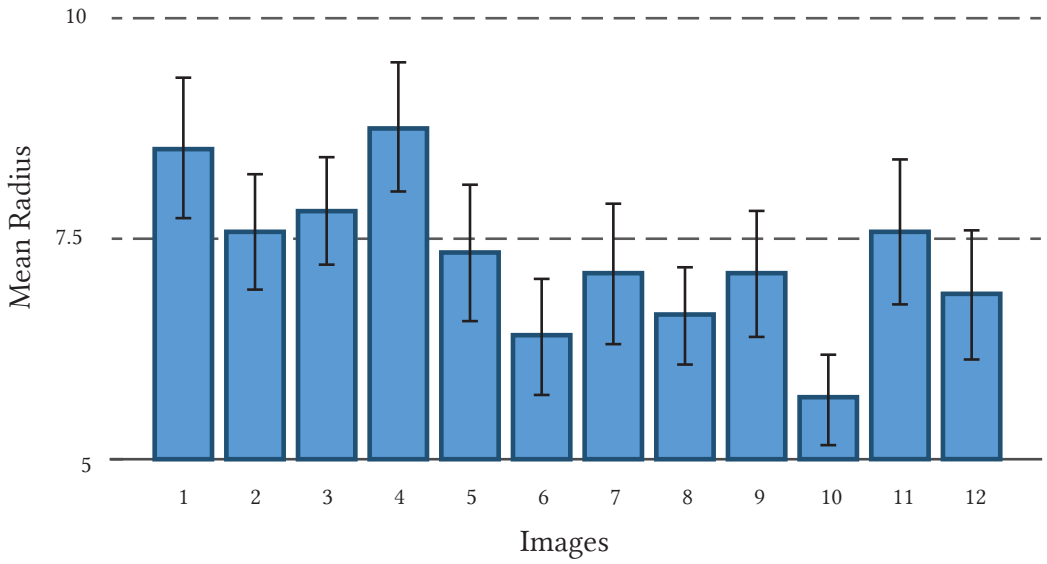


Figure 6.14. Average foveal region size preferences of the participants for each image for globally adaptive foveation. We observe a high variability between the images and we attribute this to the role of content in different images. This data shows that a traditional foveated rendering with a fixed foveal region size would not provide the optimal perceived quality. The error bars represent standard error.

measurements, our procedure was sufficient to obtain a good model, which is confirmed in our validation.

With our current implementation, it is possible to reach a running time below 1 ms for computing the prediction from our model. Nevertheless, there is still room for improvement by using lower-level optimizations, especially for Laplacian pyramid decomposition, which is the most costly operation in the implementation. We believe that an implementation which is fully optimized for the hardware would achieve much shorter running times.

It is known that the total system latency, which is mainly determined by the display refresh rate and the eye tracker sampling rate, should be taken into account when testing novel foveated rendering techniques [Albert et al., 2017]. During our validation studies, our participants have not reported any artifacts (such as so-called “popping” effects or tunnel-vision) that could be directly attributed to the system latency. However, similar to other foveation techniques, we believe that our method would also require a less aggressive level of foveation in the presence of a noticeable amount of system latency unless a countermea-

sure, such as saccade landing position prediction, is implemented (Chapter 4, Chapter 5).

Our model is currently targeting saving the computation by limiting the shading computation. We do not consider geometry pass which can have a considerable contribution to the overall rendering time. We believe, however, that our way of deriving the model, in particular, the experimental procedure and modeling, can be successfully used in the future to design more complete models for driving foveated rendering.

6.8 Conclusion

Recently proposed foveated rendering techniques use fixed, usually manually tuned parameters to define the rate of quality degradation for peripheral vision. As shown in this work, the optimal degradation that maximizes computational benefits but remains unnoticed depends on underlying content. Consequently, the fix foveation has to be conservative and in many cases, its performance benefits remain suboptimal. To address this problem, we presented a computational model for predicting a spatially-varying quality degradation for foveated rendering that remains unnoticed when compared to non-foveated rendering. The prediction is based on previous findings from human visual perception, but it is redesigned and optimized to meet the high-performance requirements of novel display technologies such as virtual reality headsets. In particular, a critical feature of our model is its capability of providing an accurate prediction based on a low-resolution approximation of a frame. Our validation experiments confirm that our technique is capable of predicting foveation which remains just-below the visibility level. This, in turn, enables optimal performance gains.

Chapter 7

Conclusion

In this thesis, we addressed two major concerns related to foveated rendering: system latency and sub-optimal performance in terms of possible computational savings.

We first focused on the latency, where there is a large discrepancy between actual gaze location and the estimation used by the rendering pipleline during rapid eye movements. In the context of foveated rendering, it means the observed actually notices with their central vision the quality degradation that should be reserved only for the periphery. To combat system latency, we presented a measurement-driven model for saccade landing position prediction. The model delivers a landing prediction with the onset of the saccade, which is then further refined with the advance of the saccade. We argued that, by utilizing the saccadic suppression, the predicted landing position can be used in the foveated rendering pipeline instead of the estimated gaze location to produce the current frame. We supported our claims by conducting two user experiments - an objective and a subjective one, both of which demonstrated the effectiveness of our model. In the first experiment, participants were asked to solve a task, whereas in the second they had to state their preference. The computational overhead of our method is negligible and it is simple to implement and integrate. A limitation of our approach is the data acquisition step: For best results an extensive amount of saccades had to be recorded from each participants causing eye fatigue for some of them.

Furthermore, we incorporated the idiosyncrasies of the saccades into our model. Considering not only between-subject variability, we also investigated how the other saccade characteristics, such as orientation, depth change, and relation to smooth pursuit eye motion, affect the saccade profile. To construct reliable models for each saccade type, tailored for each individual separately, would

require an exhausting data collection step. Therefore, we proposed a shearing operator which can modify a pre-existing dataset to match the characteristics of a certain saccade category with just a fraction of the number of saccades, required to construct a new set. We demonstrated that our saccade landing prediction model benefits from this alteration and speculated that other models, more heavily dependent on the amount of collected data, might also benefit from the shearing operator in order to be tailored for specific type of saccades.

In our work we also focus on the performance of the foveated rendering. While designing the peripheral visual acuity fall-off as a function of eccentricity is a simple and efficient rendering optimization, it is sub-optimal in its performance. The choice of the quality reduction towards the periphery lies between a conservative one, where the degradation is mild and remains unnoticeable to the observer but does not lead to high computational savings, and an aggressive one, where the savings are considerable but the observer becomes aware of the lower resolution displayed in their periphery. We proposed a predictor for content-dependent quality degradation for foveated rendering, which utilizes our knowledge of the human visual system. The model is designed to work on a low-resolution frame and it comes at a small price of additional system overhead. We demonstrated in our validation experiments that our predictor allows for significant computational savings while quality changes remain unnoticeable to the observer. We also demonstrated that our method is preferred over the standard foveated rendering when computation budget was fixed.

While our models for enhanced gaze-contingent rendering were designed after rigorous research, they are not without limitations. We already mentioned the substantial dataset required for our saccade landing position prediction. We partially addressed this issue in our later work but it remains to be verified how well our dataset performs for various eye trackers and display devices and whether acquisition of new data would be required. Currently, our prediction model is computed, and also modified, offline, before it is used. We speculated that we could perform our shear operator on the fly, while the user is interacting with the system, however, we did not verify our speculations in a perceptual experiment. No components of the underlying image are taken into consideration when performing the prediction. Similarly, our predictor for content-dependent quality degradation also relies on a massive collection of data to be properly calibrated, therefore, we resolved to using data collection technique that may have influenced the obtained calibration parameters. Vignetting and pixelasation may influence the image perception in the head-mounted displays in a way that was not measured in the scope of this work. Also, the performance of the predictor may benefit from a hardware-targeted implementation. We believe that the

aforementioned drawbacks can be successfully addressed and resolved in future works.

In this thesis we introduced the saccade landing position prediction and the luminance-contrast-aware foveated rendering separately. It is, however, possible, that in the future the two predictors might combine their strengths. The possibility of using saliency map to predict the saccade landing position prediction has already been discussed in the literature. While our luminance-contrast-aware model does not provide a prediction about regions of interest, it is possible that features from it could be extracted to serve as a "snapping" guidelines for the saccade landing prediction. In reverse, the quality degradation map could be further guided during a saccade, depending on the estimation of the prediction accuracy and the strength of the saccadic supression. A step further would be to properly integrate the models into the virtual environment. Vignetting, pixelasation, lens distortion and other factors, specific for the head-mounted displays need to be taken into greater consideration and incorporated into the perceptual models. We deem such projects as feasible in the future and that our methods could be further developed and optimized to serve the community in even more ways than the one narrated in this thesis.

While there is always room for future improvements, we believe that, within the scope of this dissertation, we have unequivocally demonstrated the usefulness and the effectiveness of our perceptually-driven models for gaze-contingent rendering.

Bibliography

Aguiar e Oliveira Junior, H., Ingber, L., Petraglia, A., Rembold Petraglia, M. and Augusta Soares Machado, M. [2012]. *Adaptive Simulated Annealing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 33–62.

Albert, R., Patney, A., Luebke, D. and Kim, J. [2017]. Latency requirements for foveated rendering in virtual reality, *ACM Trans. on App. Perception (TAP)* **14**(4): 25.

Andersson, R., Larsson, L., Holmqvist, K., Stridh, M. and Nyström, M. [2016]. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms, *Behavior Research Methods* .

Anliker, J. [1976]. Eye movement: On-line measurement, analysis, and control, *R. A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes* pp. 185–202.

Bahill, A. T., Adler, D. and Stark, L. [1975]. Most naturally occurring human saccades have magnitudes of 15 degrees or less., *Investigative Ophthalmology & Visual Science* **14**(6): 468–469.

Bahill, A. T., Clark, M. R. and Stark, L. [1975a]. The main sequence, a tool for studying human eye movements, *Mathematical biosciences* **24**(3-4): 191–204.

Bahill, A. T., Clark, M. R. and Stark, L. [1975b]. The main sequence, a tool for studying human eye movements, *Mathematical Biosciences* **24**(3-4): 191–204.

Baker, J. T., Harper, T. M. and Snyder, L. H. [2003]. Spatial memory following shifts of gaze. i. saccades to memorized world-fixed and gaze-fixed targets, *Journal of neurophysiology* **89**(5): 2564–2576.

Barten, P. G. [1989]. The square root integral (sqri): a new metric to describe the effect of various display parameters on perceived image quality, *Human*

Vision, Visual Processing, and Digital Display, Vol. 1077, Int. Soc. for Optics and Photonics, pp. 73–83.

Barten, P. G. [1999]. *Contrast sensitivity of the human eye and its effects on image quality*, Vol. 72, SPIE press.

Becker, W. and Jürgens, R. [1979]. An analysis of the saccadic system by means of double step stimuli, *Vision Research* **19**(9): 967–983.
URL: *http://www.sciencedirect.com/science/article/pii/0042698979902220*

Beeler Jr, G. W. [1967]. Visual threshold changes resulting from spontaneous saccadic eye movements, *Vision research* **7**(9-10): 769–775.

Binda, P. and Morrone, M. C. [2018]. Vision during saccadic eye movements, *Annual review of vision science* **4**: 193–213.

Boghen, D., Troost, B., Daroff, R., Dell'Osso, L. and Birkett, J. [1974a]. Velocity characteristics of normal human saccades, *Investigative Ophthalmology & Visual Science* **13**(8): 619–623.

Boghen, D., Troost, B., Daroff, R., Dell'Osso, L. and Birkett, J. [1974b]. Velocity characteristics of normal human saccades, *Invest Ophthalmology & Vis Science* **13**(8): 619–623.

Bolin, M. and Meyer, G. [1998]. A perceptually based adaptive sampling algorithm, *Proc. of SIGGRAPH*, pp. 299–310.

Bollen, E., Bax, J., Van Dijk, J., Koning, M., Bos, J., Kramer, C. and Van Der Velde, E. [1993]. Variability of the main sequence, *Invest Ophthalmology & Vis Science* **34**(13): 3700–3704.

Borji, A. and Itti, L. [2013]. State-of-the-art in visual attention modeling, *IEEE PAMI* **35**(1): 185–207.

Bouman, M. A. [1965]. Cortical control of eye movements and visual threshold., *Technical report*, Institute for perception rvo-tno soesterberg (Netherlands).

Bradley, C., Abrams, J. and Geisler, W. S. [2014]. Retina-v1 model of detectability across the visual field, *Journal of Vision* **14**(12): 22.

Bremmer, F., Kubischik, M., Hoffmann, K.-P. and Krekelberg, B. [2009]. Neural dynamics of saccadic suppression, *Journal of Neuroscience* **29**(40): 12374–12383.

Burr, D. C., Morrone, M. C. and Ross, J. [1994]. Selective suppression of the magnocellular visual pathway during saccadic eye movements, *Nature* **371**(6497): 511–513.

Burt, P. and Adelson, E. [1983]. The laplacian pyramid as a compact image code, *IEEE Trans. on Communications* **31**(4): 532–540.

Campbell, F. W. and Wurtz, R. H. [1978]. Saccadic omission: why we do not see a grey-out during a saccadic eye movement, *Vision research* **18**(10): 1297–1303.

Carpenter, R. H. [1988]. *Movements of the Eyes, 2nd Rev*, Pion Limited, UK.

Castet, E. and Masson, G. S. [2000]. Motion perception during saccadic eye movements, *Nature neuroscience* **3**(2): 177–183.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. [2014]. Describing textures in the wild, *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*.

Collewijn, H., Erkelens, C. J. and Steinman, R. M. [1988a]. Binocular coordination of human horizontal saccadic eye movements., *The Journal of Physiology* **404**(1): 157–182.
URL: *https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1988.sp017284*

Collewijn, H., Erkelens, C. J. and Steinman, R. M. [1997]. Trajectories of the human binocular fixation point during conjugate and non-conjugate gaze-shifts, *Vision research* **37**(8): 1049–1069.

Collewijn, H., Erkelens, C. and Steinman, R. [1988b]. Binocular co-ordination of human vertical saccadic eye movements, *The Journal of physiology* **404**: 183–97.

Costela, F. M. and Woods, R. L. [2019]. When watching video, many saccades are curved and deviate from a velocity profile model, *Frontiers in neuroscience* **12**: 960.

Cowey, A. and Rolls, E. T. [1974]. Human cortical magnification factor and its relation to visual acuity, *Exp Brain Res* **21**(5): 447–454.

Curcio, C. A. and Allen, K. A. [1990]. Topography of ganglion cells in human retina, *The Journal of Comparative Neurology* **300**(1): 5–25.

Daly, S. J. [1998]. Engineering observations from spatiovelocity and spatiotemporal visual models, *Photonics West'98 Electronic Imaging*, International Society for Optics and Photonics, pp. 180–191.

Deubel, H., Elsner, T. and Hauske, G. [1987]. Saccadic eye movements and the detection of fast-moving gratings, *Biological cybernetics* **57**(1): 37–45.

Didyk, P., Ritschel, T., Eisemann, E., Myszkowski, K. and Seidel, H.-P. [2011]. A perceptual model for disparity, *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011, Vancouver)* **30**(4).

Ditchburn, R. W. [1955]. Eye-movements in relation to retinal action, *Optica Acta: International Journal of Optics* **1**(4): 171–176.

Dorr, M., Martinetz, T., Gegenfurtner, K. R. and Barth, E. [2010]. Variability of eye movements when viewing dynamic natural scenes, *Journal of Vision* **10**(10): 28.
URL: *+ http://dx.doi.org/10.1167/10.10.28*

Drewes, J., Zhu, W., Hu, Y. and Hu, X. [2014]. Smaller is better: Drift in gaze measurements due to pupil dynamics, *PloS one* **9**: e111197.

Duchowski, A. T., Bate, D., Stringfellow, P., Thakur, K., Melloy, B. J. and Gramopadhye, A. K. [2009]. On spatiochromatic visual sensitivity and peripheral color LOD management, *ACM Trans. on App. Perception* **6**(2).

Duchowski, A. T., House, D. H., Gestring, J., Wang, R. I., Krejtz, K., Krejtz, I., Mantiuk, R. and Bazyluk, B. [2014]. Reducing visual discomfort of 3D stereoscopic sisplays with gaze-contingent depth-of-field, *Proc. ACM Symp. on Appl. Perc. (SAP)*, pp. 39–46.

Durbin, J. [2017]. NVIDIA estimates VR is 20 years away from resolutions that match the human eye, `https://uploadvr.com/nvidia-estimates-20-years-away-vr-eye-quality-resolution/`. Accessed: 2019-01-10.

Enright, J. [1984]. Changes in vergence mediated by saccades., *The Journal of physiology* **350**(1): 9–31.

Enright, J. [1986]. Facilitation of vergence changes by saccades: influences of misfocused images and of disparity stimuli in man., *The Journal of physiology* **371**(1): 69–87.

Erkelens, C., Steinman, R. and Collewijn, H. [1989]. Ocular vergence under natural conditions. ii. gaze shifts between real targets differing in distance and direction, *Proceedings of the Royal Society of London. B. Biological Sciences* **236**(1285): 441–465.

Gellman, R. and Fletcher, W. [1992]. Eye position signals in human saccadic processing, *Experimental Brain Research* **89**(2): 425–434.

Gregory, R. and Cavanagh, P. [2011]. The Blind Spot, *Scholarpedia* **6**(10): 9618. revision #150975.

Griffith, H., Aziz, S. and Komogortsev, O. [2020]. Prediction of oblique saccade trajectories using learned velocity profile parameter mappings, *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, New York, NY, USA, pp. 0018–0024.

Griffith, H., Biswas, S. and Komogortsev, O. [2019]. Towards reduced latency in saccade landing position prediction using velocity profile methods, *in* K. Arai, R. Bhatia and S. Kapoor (eds), *Proceedings of the Future Technologies Conference (FTC) 2018*, Springer International Publishing, Cham, pp. 79–91.

Griffith, H. and Komogortsev, O. [2020]. A shift-based data augmentation strategy for improving saccade landing point prediction, *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Adjunct, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/3379157.3388935*

Guenter, B., Finch, M., Drucker, S., Tan, D. and Snyder, J. [2012]. Foveated 3D graphics, *ACM Trans Graph (Proc SIGGRAPH Asia)* **31**(6): 164.

Han, P., Saunders, D. R., Woods, R. L. and Luo, G. [2013a]. Trajectory prediction of saccadic eye movements using a compressed exponential model, *Journal of Vision* **13**(8): 27–27.

Han, P., Saunders, D. R., Woods, R. L. and Luo, G. [2013b]. Trajectory prediction of saccadic eye movements using a compressed exponential model, *Journal of vision* **13**(8): 27–27.

Hendrickson, A. [2005]. *Organization of the Adult Primate Fovea*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–23.

Herter, T. M. and Guitton, D. [1998]. Human head-free gaze saccades to targets flashed before gaze-pursuit are spatially accurate, *Journal of neurophysiology* **80**(5): 2785–2789.

Hoffman, D., Meraz, Z. and Turner, E. [2018]. Limits of peripheral acuity and implications for vr system design, *Journal of the Soc. for Information Display* **26**(8): 483–495.

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H. and Van de Weijer, J. [2011]. *Eye tracking: A comprehensive guide to methods and measures*, Oxford University Press.

Hooge, I., Hessels, R. and Nyström, M. [2019]. Do pupil-based binocular video eye trackers reliably measure vergence?, *Vision Research* **156**: 1–9.

Hooge, I., Holmqvist, K. and Nyström, M. [2016]. The pupil is faster than the corneal reflection (cr): Are video based pupil-cr eye trackers suitable for studying detailed dynamics of eye movements?, *Vision research* **128**: 6–18.

Hooge, I., Nyström, M., Cornelissen, T. and Holmqvist, K. [2015]. The art of braking: Post saccadic oscillations in the eye tracker signal decrease with increasing saccade size, *Vision Research* **112**.

Huff, T., Mahabadi, N. and Tadi, P. [2022]. Neuroanatomy, visual cortex, *StatPearls*, StatPearls Publishing, Treasure Island (FL).

Irving, E. L. and Lillakas, L. [2019]. Difference between vertical and horizontal saccades across the human lifespan, *Experimental eye research* **183**: 38–45.

Jacobs, D., Gallo, O., Cooper, E., Pulli, K. and Levoy, M. [2015]. Simulating the visual experience of very bright and very dark scenes, *ACM Trans Graph (TOG)* **34**(3): 25.

Jang, C., Bang, K., Moon, S., Kim, J., Lee, S. and Lee, B. [2017]. Retinal 3d: Augmented reality near-eye display via pupil-tracked light field projection on retina, *ACM Trans. on Graph.* **36**(6): 190:1–190:13.

Jaschinski, W. [2016]. Pupil size affects measures of eye position in video eye tracking: Implications for recording vergence accuracy, **9**.

Jürgens, R. and Becker, W. [1975]. Is there a linear addition of saccades and pursuit movements?, *Basic mechanisms of ocular motility and their clinical implications*, Pergamon, Oxford, pp. 525–529.

Katti, H., Rajagopal, A. K., Kankanhalli, M. and Kalpathi, R. [2014]. Online estimation of evolving human visual interest, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **11**(1): 8.

Kellnhofer, P., Didyk, P., Myszkowski, K., Hefeeda, M. M., Seidel, H.-P. and Matusik, W. [2016]. GazeStereo3D: Seamless disparity manipulations, *ACM Trans. Graph. (Proc. SIGGRAPH)* **35**(4).

Kim, J., Sun, Q., Huang, F., Wei, L., Luebke, D. and Kaufman, A. E. [2017]. Perceptual studies for foveated light field displays, *CoRR* **abs/1708.06034**. **URL:** *http://arxiv.org/abs/1708.06034*

Knöll, J., Binda, P., Morrone, M. C. and Bremmer, F. [2011]. Spatiotemporal profile of peri-saccadic contrast sensitivity, *Journal of Vision* **11**(14): 15–15. **URL:** *https://doi.org/10.1167/11.14.15*

Komogortsev, O. V. and Khan, J. I. [2007]. Kalman filtering in the design of eye-gaze-guided computer interfaces, *International Conference on Human-Computer Interaction*, pp. 679–689.

Komogortsev, O. V. and Khan, J. I. [2009]. Eye movement prediction by oculomotor plant Kalman filter with brainstem control, *Journal of Control Theory and Applications* **7**(1): 14–22.

Komogortsev, O. V., Ryu, Y. S., Koh, D. H. and Gowda, S. M. [2009]. Instantaneous saccade driven eye gaze interaction, *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, ACM, pp. 140–147.

Komogortsev, O. V., Ryu, Y. S., Marcos, S. and Koh, D. H. [2009]. Quick models for saccade amplitude prediction, *Journal of Eye Movement Research* **3**(1).

Kowler, E. [2011]. Eye movements: The past 25 years, *Vision Research* **51**(13): 1457–1483.

Latour, P. [1962]. Visual threshold during eye movements, *Vision Research* **2**(3): 261–262.

Legge, G. and Foley, J. [1980]. Contrast masking in human vision, *Journal of the Opt. Soc. of America* **70**(12): 1458–1471.

Leigh, R. J. and Zee, D. S. [2015a]. *The neurology of eye movements*, OUP USA, USA.

Leigh, R. J. and Zee, D. S. [2015b]. *The neurology of eye movements*, Vol. 90, Oxford University Press, USA.

Lesmes, L. A., Lu, Z.-L., Baek, J. and Albright, T. D. [2010]. Bayesian adaptive estimation of the contrast sensitivity function: The quick csf method, *Journal of vision* **10**(3): 17–17.

Lubin, J. [1995]. A visual discrimination model for imaging system design and development, *in* P. E. (ed.), *Vision models for target detection and recognition*, World Scientific, pp. 245–283.

Mahabadi, N. and Al Khalili, Y. [2022]. Neuroanatomy, retina, *StatPearls*, StatPearls Publishing, Treasure Island (FL).

Mannos, J. and Sakrison, D. [1974]. The effects of a visual fidelity criterion of the encoding of images, *IEEE Trans. on Information Theory* **20**(4): 525–536.

Mantiuk, R., Bazyluk, B. and Tomaszewska, A. [2011]. Gaze-dependent depth-of-field effect rendering in virtual environments, *Int Conf on Serious Games Dev & Appl*, pp. 1–12.

Mantiuk, R. K., Denes, G., Chapiro, A., Kaplanyan, A., Rufo, G., Bachy, R., Lian, T. and Patney, A. [2021]. Fovvideovdp: A visible difference predictor for wide field-of-view video, *ACM Trans. Graph.* **40**(4).
URL: *https://doi.org/10.1145/3450626.3459831*

Mantiuk, R., Kim, K. J., Rempel, A. G. and Heidrich, W. [2011]. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions, *ACM Trans. Graph. (Proc. SIGGRAPH)* .

Mauderer, M., Conte, S., Nacenta, M. A. and Vishwanath, D. [2014]. Depth perception with gaze-contingent depth of field, *Proc Human Fact in Comp Sys (CHI)*, pp. 217–226.

McKenzie, A. and Lisberger, S. [1986]. Properties of signals that determine the amplitude and direction of saccadic eye movements in monkeys, *Journal of Neurophysiology* **56**(1): 196–207.

Mercier, O., Sulai, Y., Mackenzie, K., Zannoli, M., Hillis, J., Nowrouzezahrai, D. and Lanman, D. [2017]. Fast gaze-contingent optimal decompositions for multifocal displays, *ACM Trans. on Graph.* **36**(6): 237:1–237:15.

Meyer, C. H., Lasker, A. G. and Robinson, D. A. [1985]. The upper limit of human smooth pursuit velocity, *Vision research* **25**(4): 561–563.

Morales, A., Costela, F. M., Tolosana, R. and Woods, R. L. [2018]. Saccade landing point prediction: A novel approach based on recurrent neural networks, *Proceedings of the 2018 International Conference on Machine Learning Technologies*, ICMLT '18, Association for Computing Machinery, New York, NY, USA, pp. 1–5. **URL:** *https://doi.org/10.1145/3231884.3231890*

Morales, A., Costela, F. M. and Woods, R. L. [2021]. Saccade landing point prediction based on fine-grained learning method, *IEEE Access* **9**: 52474–52484.

Nvidia [2018]. VRWorks - Variable Rate Shading (VRS) website, `https://developer.nvidia.com/vrworks/graphics/variablerateshading`. Accessed: 2019-01-09.

Nyström, M., Hooge, I. and Andersson, R. [2016]. Pupil size influences the eye-tracker signal during saccades, *Vision Research* **121**: 95–103.

Nyström, M., Hooge, I. and Holmqvist, K. [2013]. Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris, *Vision research* **92**.

Ohtsuka, K. [1994]. Properties of memory-guided saccades toward targets flashed during smooth pursuit in human subjects., *Investigative ophthalmology & visual science* **35**(2): 509–514.

Ono, H., Nakamizo, S. and Steinbach, M. J. [1978]. Nonadditivity of vergence and saccadic eye movement, *Vision research* **18**(6): 735–739.

Paeye, C., Schütz, A. C. and Gegenfurtner, K. R. [2016]. Visual reinforcement shapes eye movements in visual search, *Journal of Vision* **16**(10): 15–15.

Patney, A., Salvi, M., Kim, J., Kaplanyan, A., Wyman, C., Benty, N., Luebke, D. and Lefohn, A. [2016]. Towards foveated rendering for gaze-tracked virtual reality, *ACM Trans Graph (Proc SIGGRAPH Asia)* **35**(6): 179.

Peli, E. [1990]. Contrast in complex images, *Journal of the Opt. Soc. of America* **7**(10): 2033–2040.

Peli, E., Yang, J. and Goldstein, R. B. [1991]. Image invariance with changes in size: the role of peripheral contrast thresholds, *J. Opt. Soc. Am. A* **8**(11): 1762–1774.

Pelisson, D. and Prablanc, C. [1988]. Kinematics of centrifugal and centripetal saccadic eye movements in man, *Vision research* **28**(1): 87–94.

Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A.-S., McNamara, J. O. and Williams, S. M. [2001]. Types of Eye Movements and Their Functions, *Neuroscience. 2nd edition*, Sinauer Associates.
URL: *https://www.ncbi.nlm.nih.gov/books/NBK10991*

Ramasubramanian, M., Pattanaik, S. N. and Greenberg, D. P. [1999]. A perceptually based physical error metric for realistic image synthesis, *Proc. 26th Annual Conf. on Comp. Graphics and Interactive Techniques*, SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 73–82.

Reddy, M. [2001]. Perceptually optimized 3d graphics, *IEEE Comp. Graphics and Applications* **21**(5): 68–75.

Ritter, M. [1976]. Evidence for visual persistence during saccadic eye movements, *Psychological Research* **39**(1): 67–85.

Ronchi, L. and Molesini, G. [1975]. Depth of focus in peripheral vision, *Ophthalmic Res* **7**(3): 152–157.

Salvucci, D. D. and Goldberg, J. H. [2000]. Identifying fixations and saccades in eye-tracking protocols, *Proc. Symp. on Eye Tracking Res. and Appl. (ETRA)*, pp. 71–78.

Saunders, D. and Woods, R. [2014]. Direct measurement of the system latency of gaze-contingent displays, *Behavior Research Methods* **46**(2): 439–447.

Schlag, J., Schlag-Rey, M. and Dassonville, P. [1990]. Saccades can be aimed at the spatial location of targets flashed during pursuit, *Journal of neurophysiology* **64**(2): 575–581.

Schor, C. M. [2011]. Neural control of eye movements, *Adler's Physiology of the eye*, Saunders Elsevier, Edinburgh, UK, pp. 220–242.

Sebastian, S., Burge, J. and Geisler, W. S. [2015]. Defocus blur discrimination in natural images with natural optics, *Journal of Vision* **15**(5): 16.

Shibata, T., Kim, J., Hoffman, D. M. and Banks, M. S. [2011]. The zone of comfort: Predicting visual discomfort with stereo displays, *Journal of vision* **11**(8): 11–11.

Sipatchin, A., Wahl, S. and Rifai, K. [2021]. Eye-tracking for clinical ophthalmology with virtual reality (VR): A case study of the HTC Vive Pro Eye's Usability, *Healthcare* **9**: 180.

Smeets, J. B. and Bekkering, H. [2000]. Prediction of saccadic amplitude during smooth pursuit eye movements, *Human Movement Science* **19**(3): 275–295.

Smeets, J. B. and Hooge, I. T. [2003]. Nature of variability in saccades, *Journal of Neurophysiology* **90**(1): 12–20.

Spector, R. H. [1990]. Visual Fields, *Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd edition*, Butterworths.
URL: *https://www.ncbi.nlm.nih.gov/books/NBK220*

Stein, N., Niehorster, D. C., Watson, T., Steinicke, F., Rifai, K., Wahl, S. and Lappe, M. [2021]. A comparison of eye tracking latencies among several commercial head-mounted displays, *i-Perception* **12**(1): 2041669520983338. PMID: 33628410.
URL: *https://doi.org/10.1177/2041669520983338*

Stengel, M., Grogorick, S., Eisemann, M. and Magnor, M. [2016]. Adaptive image-space sampling for gaze-contingent real-time rendering, *Comp Graph Forum*, Vol. 35, pp. 129–139.

Sun, Q., Huang, F.-C., Kim, J., Wei, L.-Y., Luebke, D. and Kaufman, A. [2017]. Perceptually-guided foveation for light field displays, *ACM Trans. Graph.* **36**(6): 192:1–192:13.

Swafford, N. T., Iglesias-Guitian, J. A., Koniaris, C., Moon, B., Cosker, D. and Mitchell, K. [2016]. User, metric, and computational evaluation of foveated rendering methods, *Proc. ACM Symp. on Appl. Perc. (SAP)*, pp. 7–14.

Tursun, O. T., Arabadzhiyska-Koleva, E., Wernikowski, M., Mantiuk, R., Seidel, H.-P., Myszkowski, K. and Didyk, P. [2019]. Luminance-contrast-aware foveated rendering, *ACM Transactions on Graphics (TOG)* **38**(4): 1–14.

Unity3D [2018]. Official website, `https://unity3d.com/`. Accessed: 2019-01-09.

Vaidyanathan, K., Salvi, M., Toth, R., Foley, T., Akenine-Möller, T., Nilsson, J., Munkberg, J., Hasselgren, J., Sugihara, M., Clarberg, P. et al. [2014]. Coarse pixel shading, *High Performance Graphics*.

Van Opstal, A. and Van Gisbergen, J. [1987]. Skewness of saccadic velocity profiles: A unifying parameter for normal and slow saccades, *Vision Research* **27**(5): 731–745.

Volkmann, F. C. [1962]. Vision during voluntary saccadic eye movements, *JOSA* **52**(5): 571–578.

Volkmann, F. C., Riggs, L. A., White, K. D. and Moore, R. K. [1978]. Contrast sensitivity during saccadic eye movements, *Vision Research* **18**(9): 1193–1199.

Wang, B. and Ciuffreda, K. [2005]. Blur discrimination of the human eye in the near retinal periphery, *Optom Vis Sci.* **82**(1): 52–58.

Watson, A. B. [2014]. A formula for human retinal ganglion cell receptive field density as a function of visual field location, *Journal of Vision* **14**(7): 15.

Watson, A. B. and Ahumada, A. J. [2011]. Blur clarified: A review and synthesis of blur discrimination, *Journal of Vision* **11**(5): 10.

Weier, M., Stengel, M., Roth, T., Didyk, P., Eisemann, E., Eisemann, M., Grogorick, S., Hinkenjann, A., Kruijff, E., Magnor, M., Myszkowski, K. and Slusallek, P. [2017]. Perception-driven accelerated rendering, *Comp. Graphics Forum* **36**(2): 611–643.

Westheimer, G. [1954]. Mechanism of saccadic eye movements, *AMA Archives of Ophthalmology* **52**(5): 710–724.

Whitmire, E., Trutoiu, L., Cavin, R., Perek, D., Scally, B., Phillips, J. and Patel, S. [2016]. Eyecontact: Scleral coil eye tracking for virtual reality, *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, ISWC '16, ACM, New York, NY, USA, pp. 184–191.
URL: *http://doi.acm.org/10.1145/2971763.2971771*

Yang, Q., Bucci, M. P. and Kapoula, Z. [2002]. The latency of saccades, vergence, and combined eye movements in children and in adults, *Investigative Ophthalmology & Visual Science* **43**(9): 2939–2949.

Yang, Q. and Kapoula, Z. [2004]. Saccade–vergence dynamics and interaction in children and in adults, *Experimental Brain Research* **156**(2): 212–223.

Yeo, S. H., Lesmana, M., Neog, D. R. and Pai, D. K. [2012]. Eyecatch: Simulating visuomotor coordination for object interception, *ACM Transactions on Graphics (TOG)* **31**(4): 42.

Young, L. and Stark, L. [1963]. Variable feedback experiments testing a sampled data model for eye tracking movements, *IEEE Transactions on Human Factors in Electronics* **HFE-4**(1): 38–51.

Zee, D. S., Fitzgibbon, E. J. and Optican, L. M. [1992]. Saccade-vergence interactions in humans, *Journal of Neurophysiology* **68**(5): 1624–1641.

Zeng, W., Daly, S. and Lei, S. [2000]. Point-wise extended visual masking for jpeg-2000 image compression, *Proc. Int. Conf. on Image Processing*, Vol. 1, pp. 657–660.

Zeng, W., Daly, S. and Lei, S. [2001]. An overview of the visual optimization tools in JPEG 2000, *Signal Processing: Image communication Journal* **17**(1): 85–104.

Zhou, W., Chen, X. and Enderle, J. [2009]. An updated time-optimal 3rd-order linear saccadic eye plant model, *International Journal of Neural Systems* **19**(05): 309–330.

Zivotofsky, A. Z., Rottach, K. G., Averbuch-Heller, L., Kori, A. A., Thomas, C. W., Dell'Osso, L. F. and Leigh, R. J. [1996]. Saccades to remembered targets: the effects of smooth pursuit and illusory stimulus motion, *Journal of Neurophysiology* **76**(6): 3617–3632.

Zuber, B. and Stark, L. [1966]. Saccadic suppression: elevation of visual threshold associated with saccadic eye movements, *Experimental neurology* **16**(1): 65–79.