

Adaptive Image-Space Sampling for Gaze-Contingent Real-time Rendering

Michael Stengel¹, Steve Groganick¹, Martin Eisemann² & Marcus Magnor¹

¹TU Braunschweig, Computer Graphics Lab, Germany, ²TH Köln, Germany

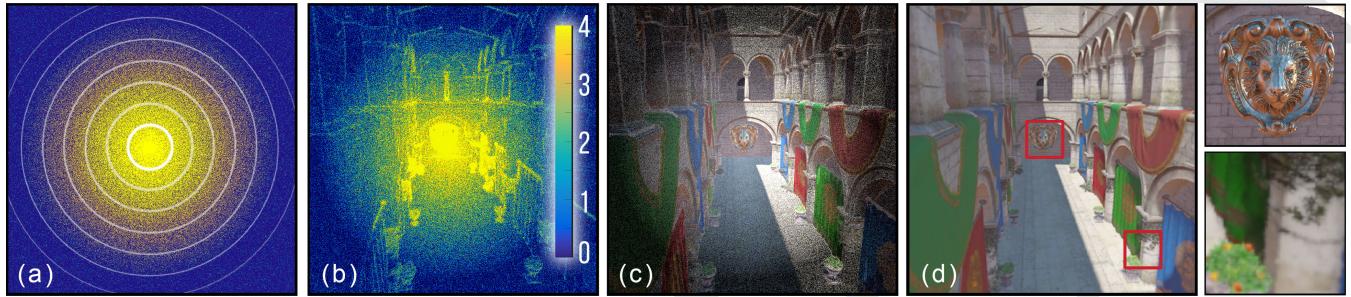


Figure 1: **Gaze-contingent Rendering Pipeline.** Incorporating visual cues such as acuity (a), eye motion, adaptation and contrast we compute a perceptually-adaptive sampling pattern (b). Sparse shading (c) and image interpolation (d) achieve the same perceived quality as shading each fragment at a fraction of the original shading costs. The resulting image contains high object detail in the foveal region (lion statue inset) and reduced detail in the periphery (flowers inset).

Abstract

With ever-increasing display resolution for wide field-of-view displays—such as head-mounted displays or 8k projectors—shading has become the major computational cost in rasterization. To reduce computational effort, we propose an algorithm that only shades visible features of the image while cost-effectively interpolating the remaining features without affecting perceived quality. In contrast to previous approaches we do not only simulate acuity falloff but also introduce a sampling scheme that incorporates multiple aspects of the human visual system: acuity, eye motion, contrast (stemming from geometry, material or lighting properties), and brightness adaptation. Our sampling scheme is incorporated into a deferred shading pipeline to shade the image’s perceptually relevant fragments while a pull-push algorithm interpolates the radiance for the rest of the image. Our approach does not impose any restrictions on the performed shading. We conduct a number of psycho-visual experiments to validate scene- and task-independence of our approach. The number of fragments that need to be shaded is reduced by 50 % to 80 %. Our algorithm scales favorably with increasing resolution and field-of-view, rendering it well-suited for head-mounted displays and wide-field-of-view projection.

Categories and Subject Descriptors (according to ACM CCS): I.3.x [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality I.3.y [Computer Graphics]: Three-Dimensional Graphics and Realism—Head-mounted Displays

1. Introduction

Modern rasterization algorithms can generate photo-realistic images. The computational cost for creating such images is mainly governed by the cost induced by shading computations. Shading has become the limiting factor in real-time rendering with ever-increasing display resolution, especially for wide field-of-view (FOV) displays such as head-mounted displays (HMD) or wide-screen projection systems.

Perceptual graphics algorithms make use of characteristics of the human visual system (HVS) to render only what we actually perceive which reduces shading computation time [RFWB07, GFD*12, HGF14, VST*14].

Our work targets gaze-contingent rendering, a subclass of perceptual graphics focusing on the exploitation of known gaze direction that is estimated by eye-tracking hardware. Several displays and novel head-mounted displays already offer integrated eye-tracking capabilities to facilitate gaze [SGE*15].

FOV of a healthy, adult human roughly extends 150° horizontally and 135° vertically per eye [AKLA11]. Most of the perceived visual information is, however, extracted from the *foveal* viewing region which measures only about 2.5° around our gaze direction [AKLA11]. Visual acuity of the HVS is highly non-uniform. Acuity is based on the number of color-sensitive cones which falls off rapidly with increasing angular deviation from the fovea (*eccentricity*). This results in less perceivable detail in our peripheral field of view. In contrast, our sensitivity to motion and temporal changes increases [Kel79]. As perception in the periphery is reduced, computation time is wasted. Specifically, assuming a uniformly resolved pixel grid a screen resolution of 9k by 8k is required to support the full FOV for a person with average vision. For next-gen HMDs companies aim for even higher pixel fill rates (16k x 16k at 240 Hz) to achieve aliasing-free low-latency VR experiences [AMD16].

Gaze-contingent rendering algorithms match rendered detail to what can actually be perceived by the user. With respect to shading, this can be easily achieved using a ray tracing framework with *selective rendering* [CDDs06] by sampling densely in the central gaze area and more sparsely towards the periphery. In rasterization, this is more challenging due to the restrictions imposed by the rasterization pipeline. GPU hardware is optimized for rendering images at one constant overall resolution. Pioneering work towards *gaze-contingent rasterization* has shown that it is capable of reducing rendering-time without affecting the perception of the displayed scene [GFD*12]. The idea is to render nested layers of increasing angular diameter and decreasing resolution which are blended to simulate the acuity fall-off.

Simplistic acuity fall-off models do not suffice to describe the perceived saliency of the HVS. Saliency is also influenced by scene content and the assigned visual task [GFD*12]. Most of the content, however, is not known until shading takes place. Therefore, one of the remaining key challenges is how to adapt render quality to saliency, not just acuity fall-off, *before* actual shading.

To address this problem, we estimate areas of high attentiveness as part of a deferred shading pipeline right after the geometry pass and before actual shading. It combines the advantages of geometry-independent deferred shading with the reduced amount of necessary detail in non-foveal vision. We model perceptual properties of the HVS to create a gaze-contingent sampling pattern. Our model incorporates a selection of prominent cues such as gaze direction, visual acuity, eye motion, areas of high contrast, and brightness. We use this pattern to locally adapt shading computation and shade only a small subset of image pixels while interpolating radiance among the remaining pixels. Our approach is implemented into a head-mounted display (HMD) setup [SGE*15].

In particular, we contribute:

- A flexible sampling scheme that is able to incorporate arbitrary perceptual cues (Sec. 4);
- An adaptive acuity model combining peripheral fall-off and eye motion (Sec. 4.1);
- A model for visual detail estimation in image-space combining spatial frequency adaptation in textures, perceptual filters in object and screen-space (Sec. 4.2), and brightness adaptation (Sec. 4.3);

- A practical, smooth multi-rate rendering scheme for perceptually lossless gaze-contingent rendering suitable for a deferred shading pipeline (Sec. 5);
- A perceptual study validating our method (Sec. 6);

Our method provides a general gaze-contingent rendering algorithm for deferred shading. It is applicable to any type of display, reduces shading cost, scales sub-linearly with image resolution and FOV and leads to significantly reduced rendering times for sophisticated high-quality shading. Its flexibility allows incorporating any perceptual cue to control sampling of the shaded pixels.

2. Related Work

Due to the vast amount of related work in the field of perceptual graphics and gaze-contingent rendering, including topics such as transmission bandwidth reduction [RLMS03, DCM04], view-dependent geometric level-of-detail [Hop98, MD01, Red01, Che03, WLC*03], or visual equivalence [RFBW07], we focus on the more closely related work. We only consider gaze-contingent approaches where the user's gaze is actively tracked [OYT96, DCM04, LU13, SGE*15] and provide an overview of two main topics that are important for the remainder of this paper: visual acuity and reduction in shading costs.

Visual Acuity describes the potential to observe details. The psychophysical model by Weymouth is a well established model for low-level vision tasks [Wey63]. The model states that the minimum discernible angular size (MAR) increases roughly linearly with eccentricity for the first 20 – 30 degrees. The angular size rises more rapidly for higher eccentricities [LKA85]. The slope of the MAR function (for simplicity described as *acuity* in the remainder of this paper) is, therefore, strongly user-dependent and cannot be precisely predicted beforehand with any model.

The simple linear model only approximates the complexity of the HVS, as peripheral vision is not a scaled down version of the foveal area [BKM05]. Eye adaptation, eye motion, accommodation and illumination influence the contrast sensitivity function (CSF) and as a result the amount of perceivable detail [Kel79, LSC04, EJGAC*15, MCNV14]. Additionally, acuity is driven by the *attentional spotlight* including mid-level vision as well as cognitive factors like faces, known objects and textures [SRJ11]. These higher-level features have so far been neglected in foveated rendering.

Selective Rendering is a flexible render method commonly used in ray-tracing to steer the number of samples per pixels or recursion depth [FPSG96, LDC06, HCS10]. Perceptual models for selective rendering simulate several properties of the HVS, but are often too costly for real-time rendering [FPSG96]. Other models for selective rendering include visual tasks and motion, as both engage the viewer's attention [YPG01, HCS10]. A drawback of selective rendering is that the techniques require an initial image estimate with at least one sample per pixel. Longhurst et al. presented a faster approach by pre-rendering a low resolution frame to extract saliency including different visual cues such as edges, intensity, motion, depth, color contrast and scene habituation [LDC06].

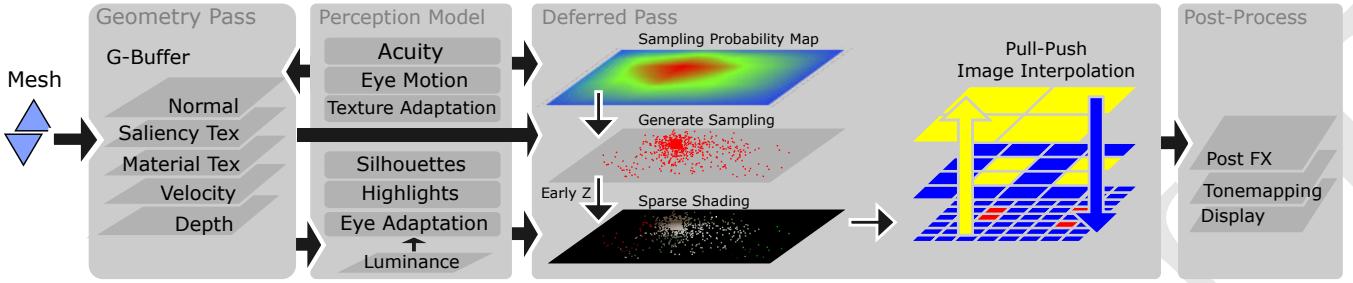


Figure 2: Overview. We describe our method for a typical deferred shading pipeline. The geometry pass generates a G-Buffer in full resolution. The G-Buffer data combined with predicted luminance, pre-computed object-based saliency and our visual perception model allows creation of a sampling probability map. In the deferred pass a sampling pattern is generated from the probability map which is then used for sparsely shading the image. Pre-processed material textures enable adapting spatial texture detail to visual acuity. A layered pull-push operation efficiently completes the missing parts of the image by interpolation. In the last step, post-processing operations like tone mapping and grading are applied before displaying the final image.

Multi-rate and Multi-resolution Shading are novel strategies in real-time applications to enable adaptation of shading complexity to the scene content.

NVIDIA recently proposed *Multi-resolution rendering* for shading at different resolutions within a single pass on their newest GPU hardware [NR15]. The approach exploits the inevitable image distortion in HMD setups and draws the image in two different resolutions subdivided into a fixed 3×3 grid. The reduced resolution saves between 20% to 50% of the pixel shading costs. However, multi-resolution rendering requires special multi projection GPU functionality whereas our approach is applicable to any GPU.

Foveated 3D graphics (F3D) [GFD*12] simulates the acuity fall-off by rendering three nested layers of increasing angular diameter and decreasing resolution around the gaze direction. These layers are blended for the final image. Unfortunately, F3D requires rasterization of the scene for each nested layer.

The idea of adaptive *Multi-rate shading* (MRS) [HGF14] is to distribute more shading samples near object silhouettes, shadow edges and regions of potential specular highlights. In blurred regions induced by motion blur or depth of field shading samples are distributed more sparsely. The approach achieves impressive savings in terms of shaded fragments (50% to 80%) without reducing perceived render quality. Efficient implementation of this approach would require an extension of the graphics pipeline which is currently not available on commodity graphics hardware.

Coarse pixel shading (CPS) [VST*14] enables different shading resolutions by executing shaders at three varying rates: per pixel group, per pixel and per sample. Results within a software renderer show shading savings comparable to MRS and applicability to foveated rendering.

3. Overview

Examples for suitable gaze-contingent rendering systems are head-mounted displays or power walls with integrated eye tracking which convey a large field-of-view. The goal of our approach is to make use of shortcomings of the HVS to determine and shade only the visually important pixels of a rasterized image. Quick interpo-

lation of color values for the remaining pixels reduces shading cost and overall rendering time. The core of our approach is deriving a per-pixel probability function \mathbf{P} for each frame, to decide which pixels should be shaded and which can safely be interpolated. This process takes place *before* actual shading (Fig. 2).

As input we make use of information available from the geometry pass, such as depth, normal, texture properties, etc. that is usually computed in modern rasterization pipelines such as Deferred Shading [ST90] or Forward+ [HMY13]. We also assume that the gaze-position \mathbf{g} is known.

First, we describe the features we incorporated into our visual perception model for sample selection and how they are combined for the final probability map \mathbf{P} (Sec. 4). We then describe how we implement the image synthesis step into a rasterization pipeline, including creation of \mathbf{P} , sampling of \mathbf{P} , shading of the selected samples, and interpolation of pixel color values for the final image (Sec. 5). We conducted a perceptual study to validate the effectiveness of our approach (Sec. 6), performed several experiments to analyze its efficiency (Sec. 7), and discuss strengths and weaknesses (Sec. 8) before concluding the paper (Sec. 9).

4. Visual Perception Model

In the following we describe our extensible model to evaluate perceptual information content of an image to create the sample probability map \mathbf{P} . The process is performed for each eye separately. Due to the real-time constraints, we consider only quickly computable features. All computations are performed in the image space of the virtual camera before performing any optional image distortion for the particular output device. The goal is to derive a per-pixel sample probability map \mathbf{P} that assigns a single *perceptual importance value* in the range $[0, 1]$ to each output pixel based on a variety of quickly computable features \mathbf{F}_0 to \mathbf{F}_n . Higher values mean higher probability for correct per-pixel shading whereas lower values indicate to use an approximate shading, e.g., by fast color interpolation.

In the following, we describe each feature \mathbf{F}_i included into our model separately, distinguishing between acuity-based (Sec. 4.1), attention-drawing (Sec. 4.2) features and global features (Sec. 4.3), and finally combine them into a single sample probability map \mathbf{P} .

4.1. Visual Acuity

Visual acuity provides an estimate of the smallest visual detail the HVS is spatially able to perceive. In our model we consider three related sub-features: Acuity fall-off, eye motion and brightness adaptation.

4.1.1. Acuity Fall-Off

Weymouth has shown an approximately linear degradation behavior of acuity with eccentricity [Wey63]. Even though this model was proven to be valid only for up to 30° of eccentricity, it is often used in methods for foveated rendering [GFD*12, VST*14, SMI16].

In our approach we extend this model to make it more suitable for use with a wide FOV. We assume a constant acuity ω_p in the far periphery as little is known currently about the acuity fall-off for eccentricities of more than 30° .

The sampling probability F_ω at a pixel position \mathbf{p} with fixated gaze position \mathbf{g} is then computed as:

$$F_\omega(\mathbf{p}, \mathbf{g}) = \text{clamp}(f(\mathbf{p}, \mathbf{g}), \omega_p, 1), \text{ with} \quad (1)$$

$$f(\mathbf{p}, \mathbf{g}) = \omega_0 + m \cdot e(\mathbf{p}, \mathbf{g}), \quad (2)$$

where $\text{clamp}(f(\mathbf{p}, \mathbf{g}), \omega_p, 1)$ clamps the values of parameter f to the range $[\omega_p, 1]$. The acuity limit ω_0 and acuity slope m are user-dependent properties which need to be computed beforehand in a calibration step (Sec. 6). The function $e(\mathbf{p}, \mathbf{g})$ computes eccentricity e in angular values. Eccentricity e is derived given the values for horizontal and vertical display resolution $d = (w, h)$, horizontal and vertical field of view $a = (\text{FOV}_h, \text{FOV}_v)$, and values for gaze position \mathbf{g} and pixel position \mathbf{p} :

$$\begin{aligned} s &= (d/2) / \tan(a/2) \\ p &= \text{atan}(|\mathbf{p} - (d/2)| / s) \\ g &= \text{atan}(|\mathbf{g} - (d/2)| / s) \\ e &= \sqrt{(p_x - g_x)^2 + (p_y - g_y)^2} \cdot (180/\pi). \end{aligned} \quad (3)$$

An example of the resultant feature map is shown in Fig. 3a. Next, we incorporate eye motion into the acuity model.

4.1.2. Eye Motion

We have described how to compute F_ω under the assumption of a static gaze per frame which is reasonable for high-refresh rate displays [GFD*12, SMI16]. The unconsciously triggered eye motion when a moving object attracts gaze, called *Smooth pursuit eye motion*, however, can invalidate this assumption for lower-refresh rates and fast gaze changes. To take eye motion into account we incorporate two sub-features: anisotropic scaling of the foveal region and motion dependent acuity adaptation.

Anisotropic Acuity Fall-off To take the expected change of the gaze position into account during frame duration Δt , we model the gaze position not as a point but as a line $\tilde{\mathbf{g}} = \mathbf{g}_i + \Delta t \lambda \tilde{\mathbf{g}}_{i-1} \mathbf{g}_i$, $\lambda \in [0, 1]$ where \mathbf{g}_i and \mathbf{g}_{i-1} are gaze positions from the current and previous frame. We compute the foveal area and acuity fall-off as described in Eq. (1) with the difference that in this case eccentricity e depends on the distance to a line instead of a position (Fig. 3b).

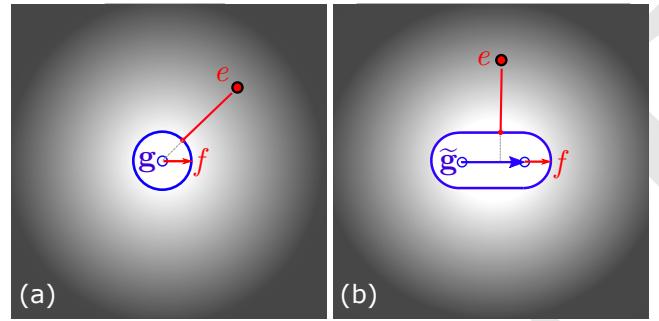


Figure 3: **Acuity-contingent sampling.** Sampling probability is 1 in the fovea and decreases with foveal distance in the periphery. Equal foveal distances result in equal sampling probabilities. **Isotropic acuity fall-off:** During eye fixations we define the foveal region by gaze vector \mathbf{g} and foveal radius f . The periphery is described by the foveal distance e . **Anisotropic acuity fall-off:** For smooth pursuit eye movement (blue arrow) the foveal region is linearly extended accordingly to the gaze motion vector $\tilde{\mathbf{g}}$.

Motion-dependent Acuity Adaptation Compensating for higher latency displays requires to increase the foveal area, but sensitivity to detail in the HVS varies with respect to velocity across the retina [Kel79]. For example, in areas where the projected velocity of the displayed object deviates from the eye motion in screen-space the sample count can be reduced. We adapt acuity computation from [Red01] to compute the motion-dependent sampling probability F_M based on the gaze motion $\tilde{\mathbf{g}}$ for any pixel \mathbf{p} :

$$F_M(\mathbf{p}, \tilde{\mathbf{g}}) = F_\omega(\mathbf{p}, \tilde{\mathbf{g}}) \cdot \mathbf{G}(\tilde{\mathbf{g}}/\Delta t), \quad (4)$$

$$\mathbf{G}(\mathbf{v}) = \begin{cases} 1.0 & \text{if } \mathbf{v} \leq 0.83^\circ/s \\ 0.002 & \text{if } \mathbf{v} > 118^\circ/s \\ 0.962 - 0.463 \log_{10}(\mathbf{v}) & \text{else.} \end{cases} \quad (5)$$

4.2. Visual Detail

In this part we introduce features that influence the user's attention and consequently should not be missed during the sampling. We also describe how to avoid visual attention-drawing artifacts stemming from careless subsampling in the peripheral viewing area. Attention and gaze do not necessarily coincide. This is known as the concepts of foveal and attentional spotlights [AKLA11]. Visual information is constantly processed in the periphery [SRJ11]. Different visual factors attract gaze more than others, specifically, regions of spatial and temporal contrast as well as saturated colors [YPG01]. It is therefore important to faithfully represent these factors in the rendered image especially in the peripheral area even though acuity is lower.

Because of this, we choose a combined approach that

1. adapts spatial texture frequencies according to perceivable detail (Sec. 4.2.1),
2. extracts prominent scene geometry features on a basis of a set of *perceptual filters* (Sec. 4.2.2).

This strategy is conservative in a way that it includes most of the

existing high frequency details in the scene in order to avoid flickering from subsampling but reduces sampling probability in areas of *visual indistinctiveness*.

4.2.1. Texture Adaptation

Textures generally represent material properties and surface details which can potentially draw the user's attention. In Sec. 4.2.2 we show how to detect these details in the object textures and thereby influence the sampling probability in \mathbf{P} .

In real-time rendering prefiltered textures, mostly mipmaps, are used to remove higher frequencies in the textures based on the projected size of the texture in image space to avoid aliasing. We extend this approach to incorporate also *resolvable detail* of the HVS, i.e. we remove texture details projected in the peripheral area. During mipmap creation we filter each level using a Gaussian filter before subsampling the texture for the next level, thus, ensuring a reduction in frequencies contained per mipmap level.

During rendering we select the mipmap level as follows: In screen space we compute the projected texel size t_s and the corresponding solid angle t_{ang} in the user's view. This value is then compared with the resolvable detail of the corresponding pixel encoded in the acuity function $\mathbf{F}_M(\mathbf{p}, \tilde{\mathbf{g}})$. In case the acuity value is higher than the angular texel size the lowest mipmap level is proposed ($l = 0$). Otherwise the mipmap level l is derived as follows:

$$l = \text{clamp}(\log_2(t_{ang}/\mathbf{F}_M(\mathbf{p}, \tilde{\mathbf{g}})), 0, \# \text{mipmaplevel}). \quad (6)$$

This value is then compared with the traditionally computed mipmap level based on the projected size and the maximum of both is taken for the final lookup.

4.2.2. Perceptual Filters

In the following we describe the usage of three different filters to retrieve potential image regions of high contrast that are perceptually significant according to previous saliency and psychophysical perception literature. We explain computational aspects of each filter in the implementation part (Sec. 5). Please note that our selection of detectors does not compromise a complete model for human perception simulation which is still an active research topic. Other perceptual as well as attentional cues affecting perception can be added easily to our model. The max-operation in Eq.12 allows selecting a fragment for rendering if any of the detectors has a positive response whereas the scaling term allows to include attributes inhibiting perceptual importance.

Object Saliency Detection Besides the usual object textures used for rendering at the same resolution we create additional object saliency textures r_{Obj} for each mipmap level. Each texel in r_{Obj} corresponds to the perceptual importance of a point on the object's surface as follows:

$$r_{Obj} = \max(\nabla r_{Norm}, \nabla r_{Bump}, \nabla r_{Alb}, r_{Gloss}, r_{Met}). \quad (7)$$

For each of the normalized material parameters albedo r_{Alb} , geometry normal r_{Norm} and detail normal r_{Bump} , we compute the maximum of the partial derivatives using a simple gradient filter of kernel size 3×3 . These gradient values are compared with the normalized metalness r_{Met} and glossiness r_{Gloss} of the material.

We take the maximum as shiny materials in physically-based rendering can easily draw the user's attention through highlights on the surface [WLC*03].

During the geometry pass the saliency textures are rendered into a separate buffer to form another feature map \mathbf{F}_O .

Silhouette Detection View-specific regions of potential contrast cannot be precomputed, as was done for the previous detector. Therefore, we employ a silhouette detection filter that works on the scene geometry from the viewpoint of the observer. The detector is essentially a gradient filter responding to changes in scene depth d and normals n . Both channels are combined by taking the maximum to form the silhouette feature map \mathbf{F}_S .

Highlight Detection Last but not least we make use of highlight detection similar to [HGF14] since our eye is sensitive to regions of high contrast [SRJ11]. This detector puts a higher sampling probability onto pixels that could potentially contain bright highlights, which could otherwise lead to flickering artifacts if not sampled properly. We compute highlight detection as:

$$\mathbf{F}_H(\mathbf{p}) = \sum_i \langle \mathbf{L}_i(\mathbf{p}), \mathbf{R}(\mathbf{p}) \rangle^\gamma \cdot \mathbf{I}_i, \quad (8)$$

where the dot product of the normalized light direction vector \mathbf{L}_i and the normalized reflection vector \mathbf{R} is scaled by the intensity \mathbf{I}_i of the i -th light source. We increase contrast using a power function with a large exponent ($\gamma = 20$) and clamp the result to zero for filtering negative light influence. Computing a dot product for each light source is inexpensive compared to full shading of the image.

4.3. Brightness Adaptation

Adaptation is the time-dependent process when the eye slowly *adjusts* to the surrounding lighting situation. The visual acuity for details and color perception are reduced with very low light intensities. On the contrary, during daytime, contrast sensitivity is lower but sharp vision and color vision work very well [LSC04, MDK08, EJGAC*15].

In our approach we follow the idea of global luminance maps [PY02] to adjust the sampling probability according to the eye adaptation. As luminance information is not available before shading, we exploit the fact that adaptation is no instantaneous effect but a process over time by using a low-frequency luminance map from the previous shaded frame I_{i-1} . Based on the RGB values before tone mapping we first compute a luminance map:

$$L(\mathbf{p}) = \langle (0.299, 0.587, 0.114)^\top, I(\mathbf{p})_{i-1} \rangle. \quad (9)$$

The exposure is iteratively updated for each frame using an empirically derived adaptation rate $a_r = 0.05$ as

$$\begin{aligned} E_i &= E_{i-1} + a_r \cdot (A - L_{avg}), \text{ with} \\ L_{avg} &= \frac{1}{N} (\sum_{\mathbf{p}} L(\mathbf{p}) \cdot \mathbf{F}_M(\mathbf{p}, \tilde{\mathbf{g}})), \end{aligned} \quad (10)$$

where E_i is the new exposure value, E_{i-1} is the previous frame's exposure, A is the user-defined auto exposure value and N is the number of pixels. Finally, I_{i-1} is tonemapped based on the adjusted exposure E_i [RSSF02] and converted again into a luminance map \bar{L} using Equation (9).

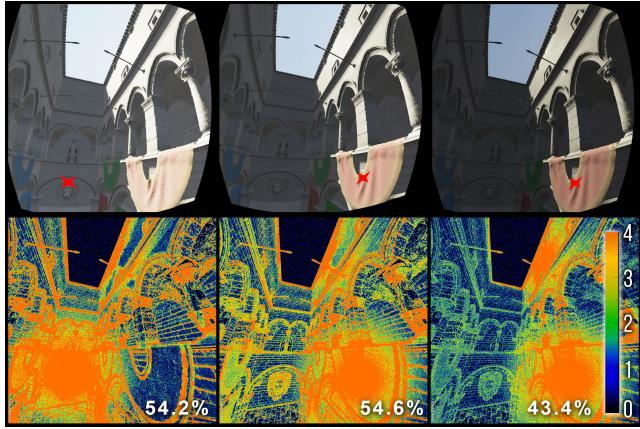


Figure 4: Brightness-adaptive sampling. Our sampling scheme distributes samples in the periphery in accordance with time-dependent adaptation. As over-exposed (left, bright wall) and under-exposed regions (right, shadow area) contain less perceivable details compared to normal exposure (center) sampling probability is reduced. Please note the relative shading count vs. per-pixel reference. The color-coding represents the number of shaded pixels in a 2×2 neighborhood according to the legend in the bottom-right.

Based on the tonemapped expected luminance distribution \bar{L} , we compute a scaling function \mathbf{S} for the per-pixel sample distribution \mathbf{P} using the following equation:

$$\mathbf{S}(\mathbf{p}) = \min\left(1, \frac{\bar{L}(\mathbf{p})}{t_{\text{dark}}}\right) \cdot \min\left(1, \frac{1 - \bar{L}(\mathbf{p})}{1 - t_{\text{bright}}}\right) \quad (11)$$

Using empirically estimated values of $t_{\text{dark}} = 0.15$ and $t_{\text{bright}} = 0.9$, the scaling function \mathbf{S} linearly scales the darkest and brightest pixels reducing sampling probability in under-exposed and over-exposed parts of the image (Fig. 4).

We finally compute the per-pixel sample probability map as the scaled maximum of all features \mathbf{F} :

$$\mathbf{P}(\mathbf{p}) = \max(\mathbf{F}_M(\mathbf{p}, \tilde{\mathbf{g}}), \mathbf{F}_O(\mathbf{p}), \mathbf{F}_H(\mathbf{p}), \mathbf{F}_S(\mathbf{p})) \cdot s(\mathbf{p}) \quad (12)$$

Hence, the probability of sampling a pixel is set according to the highest importance value returned by any of the features, which assures that visually important pixels are not missed.

5. Implementation Details

Decreasing the number of shading samples is most beneficial for high-quality render methods. For this reason we implemented our technique in a deferred rendering pipeline similar to most AAA game titles [KG13], including physically based rendering [CT82, Sch94], many-lights methods, image-based HDR environment lighting, screen-space reflections [SNRS12], HDR bloom effects, and adaptive tone mapping. An overview of the pipeline is given in Fig. 2.

We precompute the saliency textures offline (Sec. 4.2.1). In the rendering step we start with computation of the motion-compensated acuity feature map \mathbf{F}_M from gaze position and gaze motion given by the eye tracker. These are used as input to the subsequent geometry pass.

Geometry Pass In the geometry pass we render the scene from the camera view and rasterize it into the G-Buffer. The G-Buffer consists of channels for world position, normal, depth, velocity as well as material data such as albedo, roughness, metalness and cavity. For the precomputed object saliency (Sec. 4.2.2) the G-Buffer has one additional 8-bit saliency channel. We also compute shadow maps for the active light sources apart from image-based lighting.

Perceptual Probability Density Function In the second pass we evaluate the perceptual filters (Sec. 4.2.2) from the information available in the G-Buffer. Computation results of each filter are gathered in the single-channel sampling probability map.

The silhouette features detector checks for virtual edges in the scene (Fig. 5b) based on the normal and depth map of the G-Buffer (Sec. 4.2.2). Although texture normals have already been analyzed for each object in a preprocessing step, there may be regions of contrast due to object penetration or at silhouette boundaries.

Next, we include object-based saliency (Sec. 4.2.2). Since the saliency texture value depends on the motion-based acuity when being written to the G-Buffer, we now just use the texture values from the appropriate mipmap as is (Fig. 5c).

We apply highlight detection for each visible light in the scene (Fig. 5d) by rendering light meshes, evaluating the detector in the fragment shader for all affected pixels, and accumulating the results (Sec. 4.2.2). For directional lights the filter is applied on the full image by rendering a screen-aligned quad.

We estimate the brightness adaptation scaling function (Eq. 11) based on a low resolution version of the previous frame (same resolution as the acuity function texture) and combine the feature maps according to Eq. (12).

Fattening the feature map \mathbf{F} with a 5×5 dilation kernel sufficiently extends the features to avoid potential flickering in shading. To speed up this costly process we make use of a mipmap representation of the feature map and compute the dilation with a smaller kernel on a lower resolution version of the feature map \mathbf{F} .

Finally, we use the combined probability map \mathbf{P} to generate the importance sampling pattern used for shading (Fig. 5e). For every pixel \mathbf{p} we compute a pseudo-random number p [Rey98] and create a shading sample if $p < \mathbf{P}(\mathbf{p})$.

Deferred Shading Pass and Sample Interpolation Instead of saving the sampling pattern explicitly in a binary texture, we reserve the largest depth values in the depth buffer to encode the sampling pattern and scale the other depth values accordingly. Rendering a screen-filling quad using early z-culling then invokes the fragment shader only for the shading samples while efficiently discarding the rest, which was faster in our tests than discarding individual pixels in the shader. Limitations of this approach are discussed in Sec. 7.

During shading, we first accumulate shading from all primary scene lights. We then interpolate the computed radiance values as described in the following. Computation of secondary lighting effects in screen-space, such as reflections or GI, are postponed up until after interpolation.

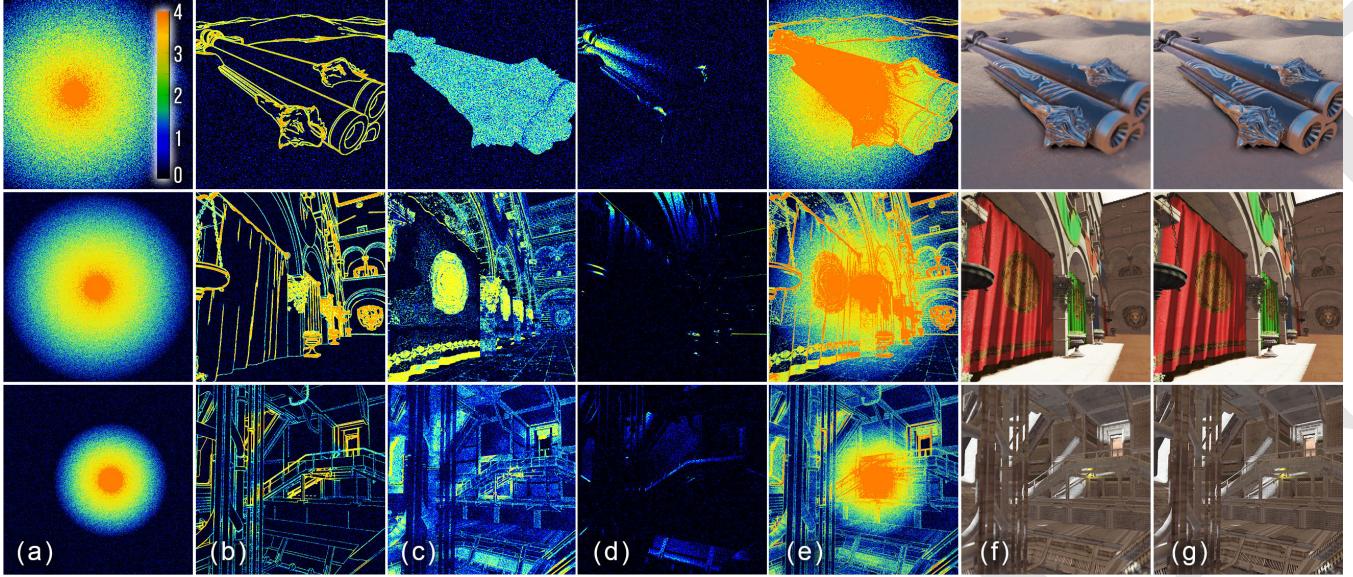


Figure 5: Sampling and shading results for three different scenes. Our renderer includes different visual cues such as visual acuity (a), silhouettes (b), object saliency (c) and specular highlights (d). The combined features allow perceptual sampling (e) for sparse shading. Pull-push interpolation fills in missing pixels resulting in a complete image (f) which is perceptually equivalent to the per-pixel shaded reference (g). The color-coding represents the number of samples in a 2×2 neighborhood.

Image interpolation is possible due to the assumption that every perceptually important detail is included in our sampling. Remaining parts can therefore be interpolated without being noticeable by the observer (Fig. 5f-g).

Edge-aware interpolation would be too costly for this step. Even the Guided Image Filter – a very efficient version of a bilateral filter – takes 30ms with an optimized CUDA implementation [BEM11]. Instead, we use a fast GPU version of pull-push for interpolation [GGSC96] based on mipmaps to fill in the missing shading information for all pixels requiring only four texture lookups per pixel in total. We can reuse the generated mipmap levels in secondary shading effects, e.g. screen-space reflections, HDR bloom effects as well as for adaptive tone mapping. Those techniques require averaged or blurred radiance values anyway [KG13, SNRS12]. Therefore, our interpolation technique does not introduce significant overhead to the render budget.

In a final step of the post processing pipeline we perform gaze-contingent temporally-adaptive tone mapping based on the averaged color information generated by the pull-push step and the updated exposure value as described earlier.

6. Perceptual Study

We conducted a perceptual study to validate that the results of our algorithm are visually equivalent to an image rendered with full per-pixel shading. Our scenes are rendered on a common desktop computer with an i7-4930K CPU and NVIDIA GTX 780 Ti graphics card with 3GB of GPU memory and displayed on a head-mounted display with integrated binocular eye-tracking. The ver-

tical field-of-view is 100° with a screen resolution of 1280×1440 pixels per eye and display refresh-rate of 60 Hz. The binocular eye-tracker gives a filtered signal at an effective output sampling rate of 75 Hz and with a precision of $\approx 1^\circ$ viewing angle. The measured latency of the eye-tracker is 12.5 ms. A worst-case latency of ≈ 50 ms may happen right after saccading eye motion before the system adjusts itself correctly again. This delay is tolerable as blur detection of the HVS does not increase significantly up to 60 ms due to post-saccadic suppression [LW07]. In our tests this latency limit is not reached in the average case.

6.1. Acuity Calibration Study

As acuity fall-off is a user-dependent property we first conducted a study to conservatively find well-working parameters for the size of the foveal region f , acuity limit ω_0 , acuity fall-off m and minimal acuity in wide periphery ω_p (Eq. 1) to avoid time-consuming calibration later on. We explained our algorithm to six participants and asked them to conservatively adjust the three parameters until they did not perceive any visual difference between a full rendering and our gaze-contingent rendering between which they could toggle at will. We presented three different test scenes (Fig. 5). For our HMD setup the following parameters have been estimated: The average acuity limit in normalized device space was $\omega_0 = 1.1715$. The linear acuity slope resulted in $m = -2.45$. The average size of the per-pixel shaded foveal region was $f = 0.07$. For the wide periphery ($e > 0.47$) the minimal acuity resulted in $\omega_p = 0.02$. The estimated conservative parameters have then been used for the second experiment with a larger group of users that is described in the next section.

6.2. Validation Study

In this study we evaluated the perceived quality of our gaze-contingent rendering by assigning different visual tasks to participants. The study was conducted with 16 persons (13 males, 3 females) who had not used the system before and not been informed about the strategy of our method. The users had normal or corrected-to-normal vision. The test started with a short introduction about our HMD and eye tracking calibration for both eyes.

The following tests were performed targeting different aspects of our proposed technique. The user had to fulfill given tasks in a virtual environment (Sponza). We performed 6 trials per test in which the respective test parameter was randomly activated or deactivated. For better comparison after each trial we switched to a gray screen with a marker to focus the user's view on the screen center again. After each pair of trials we asked the test person for perceived visual quality differences by choosing between the options "first better", "equal" or "second better".

T1 Cognitive load. The goal of this test was to draw the attention of the user to a specific and comparable task ("Count the colored spheres in the environment."). The positions of the visible spheres forced the user to look around and rotate his head in the virtual environment. The camera position automatically moved forward in the scene on a pre-defined camera track within 20 seconds. In each trial we changed the colors and positions of the spheres randomly. Each trial randomly activated either our proposed sampling method (full feature functionality) or the per-pixel shaded reference.

T2 Free viewing. In this test the task was to freely explore the environment without having a specific task which could make it easier to detect quality differences. We set a time constraint of 8 seconds. Again, each trial activated randomly either our method or ground-truth reference.

T3 Toggle manually. In this test, like in the calibration study, the user was able to toggle manually between our sampling and the reference as often as desired. This test has been performed without time constraints. Therefore, we ran the test only once per person.

T4 Brightness adaptation. In this test the user was asked to test our eye adaptation feature. In the virtual environment the user was seated in front of a wall partly lit by the sun leading to under-exposed shadow areas or over-exposed lit parts over time depending on the user's gaze position. We randomly activated or deactivated adaptive sample reduction for over- and under-exposed image regions in each trial. As before, the user was asked for visual quality differences.

T5 Eye motion. In this test we randomly activated/deactivated eye motion-based sampling to examine if the user perceives the reduced amount of detail in image parts moving differently to gaze motion. We asked the user to focus on a sphere moving into the virtual environment for 8 seconds. This moving sphere allowed us to trigger smooth pursuit eye motion that is repeatable for each trial.

T6 Texture adaptation. In the last test for each trial we randomly switched our texture adaptation feature on and off while the user was freely exploring the environment. The goal of this test was to

validate that the acuity-based peripheral textural detail reduction is not perceivable by the user.

7. Results

7.1. Shading costs

Shading Samples We tested our algorithm using three test scenes with different characteristics. The *Cerberus* scene (Fig. 5, first row) shows an old revolver in the sand which contains many specular highlights due to the shininess of the metal. The *Crytek Atrium* (Fig. 5, second row) contains complex light situations including high contrast between shadows and lit areas, or shadow edges of different intensities. The *Mothership* scene shows the engine room of a giant spaceship containing a high amount of geometric detail (Fig. 5, third row). In all cases the amount of fully shaded pixels was reduced down to roughly one third of the fragments (32.3% (*Cerberus*), 41.1% (*Sponza*), 37.2% (*Mothership*)). A detailed analysis over 300 frames is given in Fig. 6.

Rendertime The computational overhead of our method is very low, ≈ 0.9 ms for the sampling creation step for both eyes (see Sec. 6 for PC specifications and resolution), and ≈ 1 ms for the shading interpolation, which is similar to creating mipmaps. As these are needed for many effects anyway (depth of field, adaptive tone mapping, rough screen-space reflections, etc.) image interpolation comes almost for free.

Using our non-optimized prototype, we achieve for the tested scenes an overall reduction of rendertime of 25.4% on average (full rendering for both eyes requires 15.1ms per frame, our approach 11.2ms). Shading time is reduced by 41% (11.9ms to 7.0ms). The performance gain using our adaptive sampling generally depends on the shading cost per pixel. Benefits are more significant if per-pixel shading is expensive. The disparity between shaded pixels and savings in rendertime is mostly due to a hardware feature of current GPU architectures which always shade a 2×2 fragment group concurrently, no matter how many fragments are discarded [NVI16]. This limitation increases the amount of shaded pixels from 37 to 59% in average and therefore significantly reduces the shading benefit of our method on current hardware. Hopefully, next-gen GPUs remove this constraint. Integrating hardware constraints directly into the sampling procedure has not been investigated yet.

7.2. Perceptual study results

For our perceptual study a negative rating is given to all trials in which the user *correctly recognized* the subsampled image (Fig. 7, orange). In other cases the user has either seen no difference or rated the per-pixel shaded reference as being worse (Fig. 7, green).

In the passive viewing tasks (T2, T4, T6) most of the users have not been able to see a difference between our rendering variants. This validates efficiency of our perceptual model using the conservative acuity parameters derived initially. Importantly, the results of T1 and T2 don't expose any discernible difference: 87.5% (T1) and 88.9% (T2) of the users perceived no difference. Hence, our perceptual sampling performs independently from the user's cognitive load. Even in the most demanding case, T3, when comparing sampling and reference directly, the users have been able to recognize

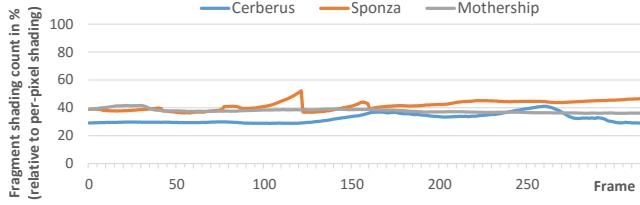


Figure 6: **Benchmark results.** For three scenes we show the amount of evoked fragment shader calls for 300 frames. Our adaptive method achieved *comparable* shading benefits in relation to the fully-shaded reference and *temporally stable* sampling rates for each test scene.



Figure 7: **Perceptual study results.** Green bars show user ratings certifying visual equivalence between our method and a full-resolution rendering in a variety of test scenarios. Orange bars show user ratings favoring the latter.

differences in visual quality only in 26.6% of the trials but phrased the differences as "unobtrusive". Some users who perceived a difference had no preference for choosing the better-looking variant.

The same holds for T6, testing our texture adaptation feature according to the acuity limit. Some users perceived the changed amount of details in the periphery (20.2%). In these cases we could increase the acuity slope resulting in more texture detail but equivalently more shaded samples. Although not being our intention, some people actually preferred the look of the "smoother-looking" image. We reason that people perceived a reduction of aliasing in the periphery which may occur in the per-pixel shaded reference.

User feedback has been very positive in T4 since the users liked the natural behavior of foveated brightness adaptation. The additional reduction of samples in washed-out image regions rarely received a negative rating (12.2%).

In T5, hardly any user (6.7% of the trials) recognized a difference between sampling the acuity fall-off only and sampling including motion-based reduction. Interestingly, the motion-based sampling reduction often reduces the number of shaded pixels to just 10 to 15% when the eye tracks moving objects or when the user moves in the environment.

8. Discussion

In the following we discuss advantages and drawbacks of our proposed pipeline and compare it with closely related approaches.

Applicability. Many modern game engines use tiled deferred shading. Lately, also the Forward+ method was proven to be very efficient for many-light scenes [HMY13]. Our method can be applied

to both approaches since both provide depth information before shading and support early z-discard to avoid expensive shading invocations. Our GPU implementation could also apply to mobile hardware with gaze-tracking, e.g. the GearVR has already been tested in combination with eye-tracking [SM16].

Our sampling technique scales favorably with resolution and FOV which are important properties for next-gen HMDs. The number of shading samples increases sublinearly with the number of pixels with increasing eccentricity whereas image interpolation scales linearly with resolution. In our test scenes the shading samples decreased from 35% on 1.2k resolution to 17% when frame resolution is doubled (2.5k per eye). Our approach provides temporal stability and predictable performance as shown in Fig. 6.

Memory Consumption. The required object saliency textures correlate to the overall texture usage in a scene. The sampling uses parts of the already available depth buffer. Besides these the memory consumption is essentially equal to standard deferred shading. The mipmap required for the interpolation is created in any case if effects like screen-space glossy reflection are incorporated.

Anti-Aliasing. Anti-aliasing techniques reduce flickering caused by undersampling of geometric and shading details. Hardware-based multisample anti-aliasing reduces the causes of geometric aliasing but cannot handle aliasing stemming from undersampling highly specular materials both spatially and temporally. Temporal anti-aliasing strategies, such as TXAA [KG14], solve this by accumulating shading information along pixel trajectories over time.

Although not yet implemented in our renderer, our adaptive sampling does not prohibit usage of TXAA. In the foveal region each pixel is shaded so that TXAA can be applied as usual. In the periphery, flickering specular highlights are avoided due to our texture adaptation (Sec. 4.2.1) which reduces frequencies in every material channel (also metalness/roughness). Pixels with specular highlights induced by small-scale geometry are sampled due to our highlight detector. In this respect, TXAA should eliminate flickering specular highlights also in the peripheral viewing areas.

In our study one user perceived peripheral flickering stemming from under-sampled shadow edges, which are not explicitly handled by our model, but could be included by the extension of He et al. [HGF14]. However, modern shadow algorithms are optimized to produce appealing soft shadows, reducing this artifact by default.

Comparison to prior work. In the following we compare our approach to the most related work of *Foveated 3D graphics (F3D)* [GFD*12], *Multi-rate shading (MRS)* [HGF14] and *Coarse pixel shading (CPS)* [VST*14].

Both MRS and CPS are currently only theoretical concepts implemented in software simulators. Both require adaptive shading features which are not available on commodity hardware. Only F3D and our approach are directly applicable to current GPUs.

Looking at the shading rates from MRS, CPS and our adaptive sampling, comparable numbers are reported with relative instruction counts of about 30 - 70 % depending on the scene complexity. This is reasonable since every previously mentioned approach shades accurately in regions of high contrast and lowers shading quality in low-contrast regions. F3D reports higher shading reductions of a factor 10–15. Several reasons cause this discrepancy:

F3D theoretically undersamples the image drastically, which results in very high frame rates. To diminish the resulting visible aliasing artifacts, F3D needs to rely on specifically designed anti-aliasing strategies, including jittered sampling of the image plane, temporal reprojection and high-refresh rates to make use of the eye integration over several frames. However, this limits F3D to simpler material models and less complex geometry. In contrast, our approach concentrates on carefully selecting samples for each single frame which allows to incorporate additional features into the visual perception model besides acuity. E.g., we provide a continuously decreasing and brightness-adaptive sampling density fall-off. We also relax the requirements of low-latency eye tracking hardware and high image refresh rates by explicitly incorporating eye motion into our sampling model and accurately selecting samples around salient geometry and material features. This increases the shading sample count but is a necessity to be more independent of specific anti-aliasing techniques or certain refresh rates.

The main goal of MRS and CPS is to evaluate the necessity to shade a pixel based on the expected image error. Once supported, these techniques would be a useful extension for our approach to further reduce shading samples in the foveal region and to adjust shading quality at shadow edges.

A novel feature, introduced in this work, from which F3D, MRS, and CPS would benefit is texture adaptation. The efficiency of MRS and CPS heavily depend on the scene content and material complexity. By reducing the material frequencies to the visual capabilities of the viewer our method is effective for arbitrary materials including bump mapping.

MRS and CPS use a regular grid at three different shading levels. F3D also uses *discrete* resolution layers. On the contrary, our sampling method may vary image quality *continuously* giving the underlying model of the HVS a much higher level of flexibility. Considering the above similarities and differences, our technique can be seen as a generalization of F3D, MRS and CPS rendering suitable for foveated rendering and commodity graphics hardware.

Future Work. Further performance improvements may be achieved by collecting material data late in the deferred pass (deferred texturing). Then, texture look-ups are only executed for those pixels that are actually shaded. In this case, the G-Buffer material data reduces to a material ID and UV coordinates instead of holding all material data.

A straight-forward idea to improve performance for gaze-contingent rendering would be view-dependent geometric level-of-detail [Red01]. Related approaches rely on drawing less vertices in the periphery or on reducing tessellation [WLC*03]. However, the HVS reacts quite sensitive to geometry changes. Therefore, an analysis of a combined method of geometry and shading adaptation rate is a promising research direction.

There is also promising work on decoupled sampling that renders defocus and motion blur with less samples by introducing a memorization cache for reusing samples across visibility samples [CTM13, RKLC*11]. Mauderer et al. show that simulation of accommodation has a positive effect on depth perception also for HMDs [MCNV14]. However, gaining performance benefits from accommodation in real-time rendering is still an open problem.

9. Conclusion

We presented a novel rendering paradigm for gaze-contingent rendering which combines the benefits of sampling flexibility (usually only available in ray tracing approaches) and fast rendering (based on a deferred shading rasterization pipeline). Our approach creates images that are *perceptually equal* to images rendered with full per-pixel shading, but at significantly reduced shading costs. For typical images with 1.2k resolution per eye our method selects only about 30-40% of the pixels for shading while interpolating the rest. For higher resolutions and wider fields-of-view, which are desired especially for HMDs, the amount is reduced down to 20% of the original pixels or less.

The approach is universally applicable in the sense that our sampling and rendering can be adapted to a variety of models of what attracts a user's attention and, what is equally important, can be computed *before* the actual shading takes place. While our approach reduces shading cost, it does not reduce the cost for the geometry pass which is rendered at full resolution. This is necessary in order to predict the visually important parts of the image. Otherwise, it would not be possible to reliably and robustly detect silhouettes or fine surface details before the actual shading.

In the future, we would like to investigate further refinements of the visual perception model. Currently, it is rather conservative as the number of samples is potentially overestimated in order to not miss any attractors. It would, however, be interesting if one could compute the *minimal* number and positions of required samples. As this is partially user-dependent, further research is required, in graphics as well as in the field of psychophysics.

Acknowledgements

We thank Andrew Maximov for providing the Cerberus asset. In addition, we thank Crytek for providing the Sponza Atrium model. We also thank Alexandre Pestana for the PBR textures originally created by Frank Meinl and Morgan McGuire. This work is part of the DFG Reinhart Koselleck Project "Immersive Digital Reality" and part of European Union's Seventh Framework Programme FP7/2007-2013 (grant no. 256941, Reality CG).

References

- [AKLA11] ADLER F. H., KAUFMAN P. L., LEVIN L. A., ALM A.: *Adler's Physiology of the Eye*. Elsevier Health Sciences, 2011. [2](#), [4](#)
- [AMD16] AMD's Raja Koduri says that we need 16k at 240hz for "true immersion" in vr, 2016. <https://t.co/0ae5Mb53WZ>, vis. 02-26-2016. [2](#)
- [BEM11] BAUSZAT P., EISEMANN M., MAGNOR M.: Guided image filtering for interactive high-quality global illumination. *Computer Graphics Forum* 30, 4 (2011), 1361–1368. [7](#)
- [BKM05] BATTISTA J., KALLONIATIS M., METHA A.: Visual function: the problem with eccentricity. *Clinical and Experimental Optometry* 88, 5 (2005), 313–321. [2](#)
- [CDdS06] CHALMERS A., DEBATTISTA K., DOS SANTOS L. P.: Selective rendering: Computing only what you see. In *4th International Conference on Computer Graphics and Interactive Techniques* (2006), GRAPHITE '06, pp. 9–18. [2](#)
- [Che03] CHENG I.: Foveated 3d model simplification. In *Signal Processing and Its Applications* (2003), vol. 1, IEEE, pp. 241–244. [2](#)

- [CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics* 1, 1 (1982), 7–24. 6
- [CTM13] CLARBERG P., TOTH R., MUNKBERG J.: A sort-based deferred shading architecture for decoupled sampling. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 141. 10
- [DCM04] DUCHOWSKI A. T., COURNIA N., MURPHY H.: Gaze-contingent displays: A review. *CyberPsychology & Behavior* 7, 6 (2004), 621–634. 2
- [EJGAC*15] E JACOBS D., GALLO O., A COOPER E., PULLI K., LEVOY M.: Simulating the visual experience of very bright and very dark scenes. *ACM Transactions on Graphics* 34, 3 (2015), 25. 2, 5
- [FPSG96] FERWERDA J. A., PATTANAIK S. N., SHIRLEY P., GREENBERG D. P.: A model of visual adaptation for realistic image synthesis. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 249–258. 2
- [GFD*12] GUENTER B., FINCH M., DRUCKER S., TAN D., SNYDER J.: Foveated 3d graphics. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 164. 1, 2, 3, 4, 9
- [GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 43–54. 7
- [HCS10] HASIC J., CHALMERS A., SIKUDOVA E.: Perceptually guided high-fidelity rendering exploiting movement bias in visual attention. *ACM Trans. Appl. Percept.* 8, 1 (2010), 6:1–6:19. 2
- [HGF14] HE Y., GU Y., FATAHALIAN K.: Extending the graphics pipeline with adaptive, multi-rate shading. *ACM Transactions on Graphics* 33, 4 (2014), Article–142. 1, 3, 5, 9
- [HMY13] HARADA T., MCKEE J., YANG J. C.: Forward+: A step toward film-style shading in realtime. *GPUPro* 4 (2013), 115–134. 3, 9
- [Hop98] HOPPE H.: Smooth view-dependent level-of-detail control and its application to terrain rendering. In *Visualization'98. Proceedings* (1998), IEEE, pp. 35–42. 2
- [Kel79] KELLY D. H.: Motion and vision. ii. stabilized spatio-temporal threshold surface. *J. Opt. Soc. Am.* 69, 10 (1979), 1340–1349. 2, 4
- [KG13] KARIS B., GAMES E.: Real shading in unreal engine 4. part of “Physically Based Shading in Theory and Practice,” *SIGGRAPH* (2013). 6, 7
- [KG14] KARIS B., GAMES E.: High quality temporal supersampling. part of “Physically Based Shading in Theory and Practice,” *SIGGRAPH* (2014). 9
- [LDC06] LONGHURST P., DEBATTISTA K., CHALMERS A.: A gpu based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th AFRIGRAPH* (2006), ACM, pp. 21–29. 2
- [LKA85] LEVI D. M., KLEIN S. A., AITSEBAOMO A.: Vernier acuity, crowding and cortical magnification. *Vision research* 25, 7 (1985), 963–977. 2
- [LSC04] LEDDA P., SANTOS L. P., CHALMERS A.: A local model of eye adaptation for high dynamic range images. In *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa* (2004), ACM, pp. 151–160. 2, 5
- [LU13] LUPU R. G., UNGUREANU F.: A survey of eye tracking methods and applications. *Bul Inst Polit Iasi* (2013), 71–86. 2
- [LW07] LOSCHKY L. C., WOLVERTON G. S.: How late can you update gaze-contingent multiresolutional displays without detection? *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 3, 4 (2007), 7. 7
- [MCNV14] MAUDERER M., CONTE S., NACENTA M. A., VISHWANATH D.: Depth perception with gaze-contingent depth of field. In *CHI'14, Proceedings* (2014), ACM. 2, 10
- [MD01] MURPHY H., DUCHOWSKI A. T.: Gaze-contingent level of detail rendering. *EuroGraphics 2001* (2001). 2
- [MDK08] MANTIUK R., DALY S., KEROFSKY L.: Display adaptive tone mapping. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 68. 5
- [NR15] NVIDIA, REED N.: Frameworks vr. *Technical Slides* (2015). 3
- [NVI16] NVidia: Life of triangle - NVIDIA’s logical pipeline, 2016. <https://goo.gl/kmHvp1>, vis. 03-21-2016. 8
- [OYT96] OHSHIMA T., YAMAMOTO H., TAMURA H.: Gaze-directed adaptive rendering for interacting with virtual space. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996* (1996), IEEE, pp. 103–110. 2
- [PY02] PATTANAIK S., YEE H.: Adaptive gain control for high dynamic range image display. In *Proceedings of the 18th spring conference on Computer graphics* (2002), ACM, pp. 83–87. 5
- [Red01] REDDY M.: Perceptually optimized 3d graphics. *IEEE Computer Graphics and Applications* 21, 5 (2001). 2, 4, 10
- [Rey98] REY J.J. W.: On generating random numbers, with help of $y = [(a + x)\sin(bx)] \bmod 1$. In *Vilnius Conference on Probability Theory and Mathematical Statistics* (1998), no. 7. 6
- [RFWB07] RAMANARAYANAN G., FERWERDA J., WALTER B., BALA K.: Visual equivalence: towards a new standard for image fidelity. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 76. 1, 2
- [RKLC*11] RAGAN-KELLEY J., LEHTINEN J., CHEN J., DOGGETT M., DURAND F.: Decoupled sampling for graphics pipelines. *ACM Transactions on Graphics (TOG)* 30, 3 (2011), 17. 10
- [RLMS03] REINGOLD E. M., LOSCHKY L. C., MCCONKIE G. W., STAMPE D. M.: Gaze-contingent multiresolutional displays: An integrative review. *HFES Journal* 45, 2 (2003), 307–328. 2
- [RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J.: Photographic tone reproduction for digital images. In *ACM Transactions on Graphics (TOG)* (2002), vol. 21, ACM, pp. 267–276. 5
- [Sch94] SCHLICK C.: An inexpensive brdf model for physically-based rendering. In *Computer graphics forum* (1994), vol. 13, pp. 233–246. 6
- [SGE*15] STENGEL M., GROGORICK S., EISEMANN M., EISEMANN E., MAGNOR M.: An affordable solution for binocular eye tracking and calibration in head-mounted displays. In *Proc. ACM Multimedia 2015* (Oct. 2015), pp. 15–24. 1, 2
- [SMI16] SMI eye tracking HMD upgrade for the Oculus Rift DK2, 2016. <http://www.smivision.com/>, vis. 03-28-2016. 4, 9
- [SNRS12] SCHERZER D., NGUYEN C. H., RITSCHEL T., SEIDEL H.-P.: Pre-convolved radiance caching. In *Computer Graphics Forum* (2012), vol. 31, Wiley, pp. 1391–1397. 6, 7
- [SRJ11] STRASBURGER H., RENTSCHLER I., JÜTTNER M.: Peripheral vision and pattern recognition: A review. *Journal of Vision* 11, 5 (2011), 13. 2, 4, 5
- [ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *ACM SIGGRAPH Computer Graphics* (1990), vol. 24, ACM, pp. 197–206. 3
- [VST*14] VAIDYANATHAN K., SALVI M., TOTH R., FOLEY T., AKENINE-MÖLLER T., NILSSON J., MUNKBERG J., HASSELGREN, ET AL.: Coarse pixel shading. In *Eurographics/ACM SIGGRAPH Symposium on High Performance Graphics* (2014), The Eurographics Association, pp. 9–18. 1, 3, 4, 9
- [Wey63] WEYMOUTH F. W.: Visual sensory units and the minimum angle of resolution. *Optometry Vision Science* 40 (1963), 550–568. 2, 4
- [WLC*03] WILLIAMS N., LUEBKE D., COHEN J. D., KELLEY M., SCHUBERT B.: Perceptually guided simplification of lit, textured meshes. In *Proceedings of the 2003 symposium on Interactive 3D graphics* (2003), ACM, pp. 113–121. 2, 5, 10
- [YPG01] YEE H., PATTANAIK S., GREENBERG D. P.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Trans. Graph.* 20, 1 (2001), 39–65. 2, 4