Perceptual Model for Adaptive Local Shading and Refresh Rate

AKSHAY JINDAL, University of Cambridge, UK
KRZYSZTOF WOLSKI, MPI Informatik, Germany
KAROL MYSZKOWSKI, MPI Informatik, Germany
RAFAŁ K. MANTIUK, University of Cambridge, UK

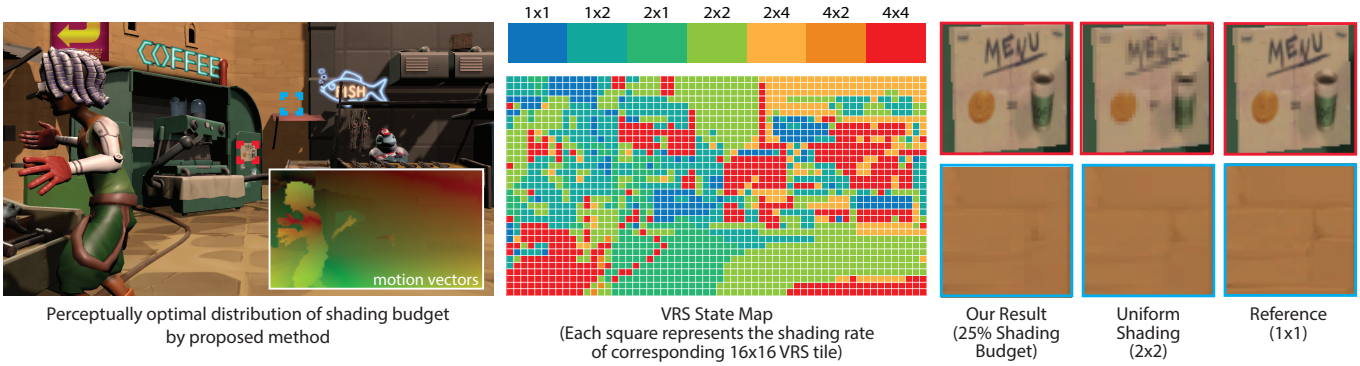


Fig. 1. We propose a novel method for adaptive control of local shading (VRS) and refresh rate, which maximizes the quality of animated content under constrained rendering budget. The method analyzes texture content, motion vectors, luminance, and angular display resolution to propose both a refresh rate and a VRS state map for a given frame. In the shown example with camera motion, which is rendered with a fixed shading budget of 25% pixels/frame, our method assigns higher shading rate to the regions where quality degradation due to variable-rate shading (VRS) is more visible (high-frequency textures, low velocities) and lower shading rates to less perceptible regions (smooth textures, high velocities).

When the rendering budget is limited by power or time, it is necessary to find the combination of rendering parameters, such as resolution and refresh rate, that could deliver the best quality. Variable-rate shading (VRS), introduced in the last generations of GPUs, enables fine control of the rendering quality, in which each 16×16 image tile can be rendered with a different ratio of shader executions. We take advantage of this capability and propose a new method for adaptive control of local shading and refresh rate. The method analyzes texture content, on-screen velocities, luminance, and effective resolution and suggests the refresh rate and a VRS state map that maximizes the quality of animated content under a limited budget. The method is based on the new content-adaptive metric of judder, aliasing, and blur, which is derived from the psychophysical models of contrast sensitivity. To calibrate and validate the metric, we gather data from literature and also collect new measurements of motion quality under variable shading rates, different velocities of motion, texture content, and display capabilities, such as refresh rate, persistence, and angular resolution. The proposed metric and adaptive shading method is implemented as a game engine plugin. Our experimental validation shows a substantial increase in preference of our method over rendering with a fixed resolution and refresh rate, and an existing motion-adaptive technique.

Authors' addresses: Akshay Jindal, University of Cambridge, UK, aj577@cst.cam.ac.uk; Krzysztof Wolski, MPI Informatik, Germany, kwolski@mpi-inf.mpg.de; Karol Myszkowski, MPI Informatik, Germany, karol@mpi-inf.mpg.de; Rafał K. Mantiuk, University of Cambridge, UK, rafal.mantiuk@cl.cam.ac.uk.

0730-0301/2021/12-ART281
https://doi.org/10.1145/3478513.3480514

CCS Concepts: • **Computing methodologies** → **Perception**; *Rasterization.*

Additional Key Words and Phrases: variable rate shading, motion quality, bandwidth-limited rendering

ACM Reference Format:
Akshay Jindal, Krzysztof Wolski, Karol Myszkowski, and Rafał K. Mantiuk. 2021. Perceptual Model for Adaptive Local Shading and Refresh Rate. *ACM Trans. Graph.* 40, 6, Article 281 (December 2021), 18 pages. https://doi.org/10.1145/3478513.3480514

1 INTRODUCTION

Shading is one of the most computationally expensive parts of the rendering pipeline, yet the demand for shading computations is growing significantly with the increase of both the resolution and refresh rate of displays. Even the most powerful GPUs are unable to meet these demands and are limited by their computational power and bandwidth. Furthermore, with the growing popularity of mobile gaming, which needs to operate under a limited power budget, and also GPU sharing cloud gaming [Yadav and Annappa 2017], rendering often needs to operate at a fraction of the maximum GPU capacity. To address this challenge, all popular GPU manufactures have introduced a more flexible shading mechanism called VRS in their next-gen chipsets [AMD 2021; Nvidia 2018; Qualcomm 2021]. VRS enables the control of the resolution of shading within each 16×16 image tile, while retaining visibility computation at the native resolution. VRS has been used to exploit the limits of the human visual system (HVS) by intelligently distributing the shading budget based on foveation [Tursun et al. 2019], scene content and motion

[Yang et al. 2019], or depth-of-field [Intel 2019]. All these works propose a dynamic quality control mechanism that allocates the rendering budget to those aspects of an image or animation that have the highest impact on the overall quality. In this work, we propose to control both the VRS state map and the refresh rate, based on all major factors affecting image quality: texture content, on-screen velocities, luminance, effective resolution, and display persistence. We build on the work of Denes et al. [2020], and extend the visual quality model to account for the important effect of texture content and VRS resolution. In contrast to that work, we offer local, rather than global, control of the resolution via VRS without the need for eye tracking.

The key component of our Adaptive Local Shading and Refresh Rate (ALSaRR) method is a new Content-adaptive Metric of Judder, Aliasing and Blur (CaMoJAB) (Section 4). The metric is based on psychophysical models of contrast sensitivity with only a few parameters fitted to the data. We calibrate and validate our metric on various existing datasets as well as our new dataset collected by conducting a subjective quality experiment. The experiment measures the perceived loss of quality due to shading rate reduction under a large range of display refresh rates, resolutions, display persistence, luminance, contrast, and content velocity (Section 4.3). Our ALSaRR method uses the new metric to create per-texture quality functions, which are used for an approximate solution of the knapsack problem: maximize perceived quality for a given rendering budget (Section 5). Unlike the method of Yang et al. [2019], which controls VRS to avoid any visual loss regardless of the per-frame rendering cost, our goal is to find the best trade-off of spatio-temporal resolution under a limited rendering budget.

The main contributions of our work[1] are:

- Content-adaptive Metric of Judder, Aliasing and Blur (CaMoJAB), derived from psychophysical models and calibrated on several datasets.
- A dataset of motion quality for animations rendered with different shading rates, motion velocities, textures, refresh rates and display angular resolution (in pixels-per-degree), and persistence.
- Adaptive Local Shading and Refresh Rate (ALSaRR) method for control of real-time rendering, which maximizes the quality of animation under a limited budget.

2 BACKGROUND

We begin with a brief overview of the landscape of motion artifacts prevalent on modern display technologies (Section 2.1). It is followed by a review of psychophysical studies exploring the effect of various display and content parameters on the perception of these artifacts (Section 2.2). Finally, we discuss a contemporary shading technology VRS that can be utilized for motion adaptive rendering (Section 2.3).

2.1 Motion artifacts on modern displays

Real-time rendering of animated content is prone to various artifacts due to software and hardware limitations. Working with frame rates lower than display refresh rates can give rise to screen *tearing* or *stuttering* artifacts. Asymmetric pixel transition time of LCD panels leads to *trailing* artifacts. Response time compensation used to mitigate trailing may lead to *coronas (glowing edges)* [Jokinen and Nivala 2017]. *Blur* is perceived whenever a tracked object has a non-zero retinal velocity. This can happen either due to the sample and hold nature of some display technologies (hold-type blur) or imperfect eye motion when tracking an object [Denes et al. 2020]. One way to reduce blur is to reduce the amount of time a signal is displayed every frame (aka persistence) but this can lead to *flicker* if the display's refresh rate is lower than the critical fusion frequency of the HVS and also from a lower accuracy of saccadic eye motion [Goettker et al. 2020]. Low refresh rates also lead to *judder*/non-continuous motion perception. On low persistence displays, such as HMDs or projectors with butterfly shutters, *ghosting/false-edges* is a common artifact that occurs when the image of a moving object is displayed multiple times at the same location [Scott Murdison et al. 2019]. Finally, the visibility of these artifacts also depends on the displayed content. When the display's spatial and temporal sampling frequency is lower than that of the displayed signal, we see *aliasing* artifacts which are a common occurrence in real-time graphics.

The artifact visibility can be suppressed by actively modifying image content. For example, increasing motion blur by means of rendering [Navarro et al. 2011; Sung et al. 2002] or longer camera exposure [Fuchs et al. 2010] can reduce aliasing, ghosting and flicker [Hoffman et al. 2011; Stengel et al. 2015]. However, the methods that alter the content are not considered in this work. Artifact suppression can also be achieved by hardware adjustments such as syncing display refresh rate with the frame rate of an application (adaptive-sync), or over-driving liquid crystals to reduce their response time [Feng 2006]. Unfortunately, achieving an artifact-free motion rendering is often an impossible task due to the current display limitations and usually involves trading off between different artifacts. In the next section, we discuss several factors that affect the perception of these artifacts and motion quality.

2.2 Factors affecting perception of motion artifacts

The perceived quality of reproduced motion is a well-studied problem with various works exploring the effect of the following display and content dependent factors on quality of motion:

Refresh-rate and velocity: There is a consensus among multiple studies examining a wide range of refresh rates and velocities (refer to Table 1) that higher refresh rates are required to maintain the quality at higher velocities of motion. Mackin et al. [2016] report that even for a simple stimulus, it can take upto 600 Hz (depending upon the velocity) to achieve an artifact-free motion. However, observers found the motion artifacts beyond 300 Hz to be tolerable. They also noticed that temporal aliasing artifacts (judder and ghosting) contribute more to motion quality impairment compared to hold-type blur. In another study, Kuroki et al. [2007] measured motion quality up to refresh rate of 480 Hz and found that the quality improves rapidly with increasing refresh rate but saturates after 240 Hz for natural images. Similar trends were also reported by [Chapiro et al. 2019; Denes et al. 2020; Wilcox et al. 2015]. Similar to temporal aliasing, the visibility of flicker decreases with increasing refresh rate. These trends stay mostly the same for stereoscopic presentation

[1] Source code and dataset at: https://www.cl.cam.ac.uk/research/rainbow/projects/alsarr

[Hoffman et al. 2011]. However, unlike temporal aliasing, flicker can be detectable up to 500 Hz during saccadic eye motion [Davis et al. 2015].

Persistence: Also known as duty-ratio or shutter angle, is the fraction of the frame duration the signal is displayed. A substantial difference in judderness can be perceived [Daly et al. 2015] between shutter angles that differ by a large amount. Furthermore, it is necessary to lower persistence to significantly reduce the hold-type blur inherent to LCDs as simply increasing the refresh rate is not effective [Sluyterman 2006]. In fact, LCD panels with strobing backlights [Feng 2006] are becoming increasingly common in head mounted displays (HMDs) and gaming monitors [Rejhon 2017]. Unfortunately, psychophysical studies on the range of persistence used in these devices are quite limited (refer to Table 1).

Luminance and contrast: The models of contrast sensitivity [Barten 2003] predict that the HVS is more sensitive to higher luminance and contrast and thus one would expect a similar trend for visibility of motion artifacts. Larimer et al. [2001] verified this in their study where they measured the effect of luminance, contrast, and refresh rate on judder and flicker up to 100 cd/m^2 and found luminance to be the dominant variable. Roberts and Wilkins [2013] noted that the visibility of flicker on a square-wave signal increases with increasing Michelson contrast and Daly et al. [2015] reported similar degradation of overall motion quality with increasing contrast.

Resolution and spatial frequency: Visibility of blur is linearly dependent on the logarithm of the averaged spatial frequency of an image [Kuroki et al. 2007] with observers showing a slightly higher preference for higher spatial frequency content [Daly et al. 2015; Navarro et al. 2011]. The content resolution has also shown to be an important factor for rendering on variable refresh rate displays with viewers picking higher refresh rates for low computational budget and higher resolution for higher budgets [Debattista et al. 2018]. Reducing the resolution in rendering results in blur and aliasing artifacts but how our sensitivity to these artifacts changes with velocity is still not well studied (Table 1).

Other factors: While there is also some evidence of effect of motion direction [Daly et al. 2015], motion predictability [Denes et al. 2020], color [Daly et al. 2015] and the type of eye motion [Roberts and Wilkins 2013] on motion quality, we do not explore them in this work.

We summarize the above discussion in Table 1 and report the range of each of the parameters measured in the existing work. We find the studies of persistence and resolution to be particularly lacking as they do not cover the ranges commonly found in VR and mobile devices. Furthermore, spatio-temporal aliasing artifacts due to low-resolution rendering are also not well considered in these works. Lastly, a majority of these studies used oversimplified stimuli such as lines, boxes, etc. that may not generalize well to complex image content. We attempt to fill these gaps in knowledge by conducting a psychophysical study on multiple display setups and more realistic texture content as described in Section 4.3. In the first column of Table 1 we mark all datasets, including ours, that we used to calibrate and validate our motion quality model (Section 4).

2.3 Variable Rate Shading

VRS [Nvidia 2018] or coarse pixel shading [Vaidyanathan et al. 2014] are recent technologies, available on the latest GPUs. VRS decouples shading and visibility calculations to provide flexible control over shading. By rasterizing at full resolution but shading at a lower resolution, the technique reduces pixel shading load while preserving edges and visibility of objects. Unlike their predecessors, Multi-Resolution Shading and Lens-Matched Shading [Kraemer 2018], which optimize shading workload to match VR optics, VRS allows for much more granular control, where every 16×16 tile of pixels can have a different shading rate. The recent specification offers seven different shading rates (1×1, 1×2, ..., 4×4) including non-uniform shading resolution along horizontal and vertical dimensions. Since VRS only executes the fragment shader a few times per multiple raster pixels and broadcasts the same shaded value to the neighboring pixels, the artifacts generated by VRS on the surface of each object are the same as upsampling with a box filter. The visibility of these artifacts depends on several factors such as luminance, refresh rate, velocity, and content, as seen in the results of our Experiment 1 (Section 4.3) where we use VRS to control the resolution of our stimuli.

3 RELATED WORK

In this section, we discuss the existing models of motion quality (Section 3.1), followed by rendering applications that motivate such metrics (Section 3.2 and Section 3.3).

3.1 Models of motion quality

Motion quality metrics aim to predict the visibility of motion artifacts given content and display information as input. Such models can then be used for various motion-adaptive rendering applications. Primarily, there are two main approaches of modeling quality: black-box metrics and white-box metrics. Black-box metrics [Chapiro et al. 2019; Debattista et al. 2018] explain the data by fitting an arbitrary function, and typically perform well within the domain of the dataset, but are unable to extrapolate beyond it. White-box metrics on the other hand rely on psychophysical models and are better at extrapolating the predictions. We focus on white-box metrics as we aim to build a metric that generalizes to a wide range of display technologies and content.

Denes et al. [2020] proposed a white-box metric (called MARRR) that models motion quality as a weighted sum of the quality of individual motion artifacts. The metric makes a simplifying assumption that motion artifacts are independent of each other which fails for content-adaptive scenarios where blur and spatio-temporal aliasing from rendering are often inter-dependent. Hoffman et al. [2011] analyses the frequency spectra of a moving line to predict the visibility of motion artifacts, however, it is unknown if such predictions will extend well to supra-threshold appearance. Both the metrics ignore the impact of content on motion quality.

Yang et al. [2019] proposed a metric for adaptive control of the VRS, called Nvidia Adaptive Shading (NAS), which considers both the screen content and the velocity. The metric analyses the screen content in the frequency domain and uses that information together with the motion vectors to classify each VRS tile into full, half, and

Table 1. Existing motion quality studies and the range of motion artifacts and display/content parameters explored by them. Our experiments focused on understanding the effect of resolution and display persistence on visibility of motion artifacts, particularly blurring and aliasing artifacts caused by VRS.

Motion Quality Datasets	Motion Artifacts						Range of Display/Content Parameters Studied						
	Judder	Flicker	Blur	Ghosting	Aliasing (Rendering)	Depth Distortion	Refresh Rate (Hz)	Persistence* (Duty Cycle)	Resolution (PPD)	Velocity (deg/s)	Adaptation Luminance (cd/m2)	Contrast (Michelson)	Stimuli
[Larimer et al. 2001]	✓	✓		✓			15 : 30	0.01 : 1	×	0.46 : 5.46	7 : 110	0.5 : 1	Vertical Bars
[Kuroki et al. 2007]	✓		✓				60 : 480	CRT	64	0 : 80	40	Images	CG+Natural Images
[Navarro et al. 2011]			✓		✓		60	1	110	5 : 15	300	×	CG Images
[Hoffman et al. 2011]	✓	✓	✓	✓		✓	10 : 50	CRT	42	1 : 16	30 : 60	×	Moving Box
[Roberts and Wilkins 2013]		✓					1000 : 5000	0.01	×	0	0.02 : 310	0.05 : 0.4	Stationary Lines
[Johnson et al. 2014]	✓	✓	✓	✓		✓	30 : 240	0.25 : 0.5	60	0 : 25	10 : 50	×	Moving Box
[Davis et al. 2015]		✓					25 : 1000	×	87	0	20 : 2700	0.09 : 0.95	Stationary Edge
[Wilcox et al. 2015]	✓		✓				24 : 60	DLP	118	Videos	16	Videos	Natural Videos
[Daly et al. 2015]	✓	✓	✓	✓			12 : 60	DLP	30	0.8 : 1.3	5 : 65	0.125 : 1	Gabor + Natural Images
[Mackin et al. 2016]§	✓		✓	✓			60 : 2000	0.01	Printed Stimulus	10 : 70	150 : 3200	×	Moving Edge
[Debattista et al. 2018]	✓		✓		✓		15 : 120	1	640 : 2560 px	3D Animation	350	Images	CG Videos
[Denes et al. 2020]§	✓		✓				15 : 165	1	62	15 : 45	36	Images	CG+Natural Images
ITU-R [Series 2020]§	✓	✓	✓	✓			60 : 240	0.3 : 0.9	7 : 60	8 : 32	169 : 480	Videos	Natural Videos
Ours§	✓		✓		✓		60 : 144	0.05 : 1	13 : 90	0 : 75	2.5 : 150	0.7 : 0.9	CG+Natural Images

* Persistence is the display's duty cycle and should not be confused with the camera's shutter angle. It is not well-defined for CRT and DLP displays due to asymmetric phosphorus decay time and use of color wheel, respectively.
× Not reported in their publication.
§ Datasets that were used to calibrate/validate our model.

quarter resolution shading. The objective of their metric is to use the lowest possible shading rate that results in visually lossless quality. Unlike our method, which keeps the computational cost the same across the frames, their method will result in varying rendering cost across the frames. Therefore, their method is more suitable for the applications in which the rendering budget is unconstrained, but it is desirable to keep it low, for example, to reduce the power consumption. A more detailed discussion of their method can be found in Section 6. Perceptual video quality metrics such as [Mantiuk et al. 2021] are also designed to be content and motion-aware but their complexity precludes real-time rendering.

Furthermore, most of the existing models only consider a small subset of motion artifacts. One artifact that is rarely considered is temporal and spatial aliasing due to rendering (Table 1). Motion artifacts are often inseparable and studying them in isolation does not translate well to practical applications. Aliasing due to rendering is also particularly important for our application as it is one of the most common artifacts resulting from shading rate reduction. The summary of the discussed metrics can be found in Table 2. Our proposed metric accounts for all of the important factors described in Section 2.2 and outperforms the above models, as demonstrated in Section 4.5.

Table 2. Models of motion quality and the parameters they account for.

Motion Quality Metrics	Refresh Rate	Resolution	Persistence	Velocity	Luminance	Content-aware
[Hoffman et al. 2011]	Yes	No	Yes	Yes	No	No
[Debattista et al. 2018]	Yes	Yes	No	No	No	No
NAS [Yang et al. 2019]	No	Yes	No	Yes	Yes	Yes
[Chapiro et al. 2019]	Yes	No	No	Yes	Yes	No
MARRR [Denes et al. 2020]	Yes	Yes	Yes	Yes	No	No
CaMoJAB (ours)	Yes	Yes	Yes	Yes	Yes	Yes

3.2 Motion adaptive rendering

Offline rendering algorithms can make use of visual models for adaptive sampling. Ray tracers can be developed to directly synthesize an image in the frequency domain [Bolin and Meyer 1995] which makes it easier to modulate them with models of visual masking [Ferwerda et al. 1997]. Visible difference predictors could be used to determine the stopping conditions of progressive Monte-Carlo rendering [Myszkowski 1998] or further degrade the quality of moving objects [Myszkowski et al. 1999; Navarro et al. 2011; Yee et al. 2001]. Other high-level visual models such as crowding [Jarabo et al. 2012], attention [Cater et al. 2003] and saliency maps [Longhurst et al. 2006] have also been successfully used in decreasing the computational cost in path tracers. An extensive survey on perceptually accelerated sampling can be found in [Weier et al. 2017]. The above techniques do not account for display aspects and involve computationally expensive quality metrics making them unsuitable for real-time rendering.

Locally adaptive real-time shading is a relatively new area as the traditional rasterization frameworks did not allow for a sub-image level of control until recently. The major focus of these developments has been on rendering for VR with the goal of either matching the non-uniform pixel distribution of HMDs [Pohl et al. 2015] or exploiting the reduction of visual acuity with eccentricity [Tursun et al. 2019]. The recent introduction of VRS allowed for controlling the shading rates of each 16×16 pixel tiles enabling a plethora of optimizations. Vaidyanathan et al. [2012] analyzed motion and defocus blur in the frequency domain to adaptively control the shading rate

and Yang et al. [2019] developed a similar framework better suited for VRS hardware. Drobot [2020] extends this idea to all platforms through their software based implementation of VRS. Schmid et al. [2019] describes how ray-tracing of reflections can be done in a fashion similar to VRS by varying the number of ray samples for each 8×8 pixel tile. Most of the above techniques are not perceptually motivated and their main focus has been on reducing the rendering cost while yielding visually equivalent output. We argue that the goal of obtaining no visual loss is an impractical target for real-time rendering, as producing high-quality results may require a much larger computational budget than available. In contrast to those works, our aim is to deliver the minimum degradation of quality under a constrained rendering budget.

3.3 Fixed-bandwidth rendering

Fixed-bandwidth rendering involves the optimal distribution of rendering resources without exceeding a fixed rendering budget given in *pixels/sec*. This can be done by either trading off between resolution and refresh rate [Debattista et al. 2018; Denes et al. 2020] or by fixing the refresh-rate and dynamically changing the resolution [Unity 2021; Unreal 2021]. All of these works control the global resolution of a frame and extending them to locally adapt the resolution is non-trivial.

The problem of maximizing quality for a given bandwidth is well studied in the rate-distortion theory but the common solutions involve coding tree units and dynamic programming [Ortega and Ramchandran 1998] not suitable for real-time rendering. Our proposed optimization is similar to Li et al. [2017] who found the optimal trade-off between delay, power, and rate-distortion in dynamic adaptive streaming applications by formulating it as an integer linear programming problem and provide a greedy approximation. We derive a similar formal bound on quality more suited for VRS and propose a parallel implementation on GPU that meets the demands of real-time rendering.

4 CAMOJAB: CONTENT-ADAPTIVE METRIC OF JUDDER, ALIASING AND BLUR

Our goal is to design a motion quality model that, given a display configuration and an image, can predict how the quality will change with display refresh rate, shading resolution, display persistence, and image velocity. We aim to make this model as simple as possible for real-time usage while covering a wide range of display parameters and motion artifacts. The texture content is a major factor influencing quality degradation due to VRS, however, most existing perceptual models ignore the effect of content on motion quality. Therefore, we also aim to make our model content-adaptive.

In the following sections, we develop a motion quality model by studying the evolution of a continuous motion signal and how it goes through various spatial and temporal distortions due to the discrete nature of rendering and display systems. We focus on the distortion that VRS introduces to the mapped texture and we assume that the distortion due to a lower resolution of shading (e.g. specular reflections) is negligible. The following analysis models an image that is stabilized on the retina therefore we do not consider the changes of the signal over time. We define the motion quality q as a negate of weighted sum of spatial distortions (d_s) and temporal distortions (d_t) and show that such a definition correlates well with the experimental results.

$$\begin{aligned} q(t(u,v), v, \boldsymbol{D}) = &- w_\mathrm{s} \cdot d_\mathrm{s}\left(t(u,v), v, \boldsymbol{D}\right) + \\ &- w_\mathrm{t} \cdot d_\mathrm{t}\left(t(u,v), v, \boldsymbol{D}\right), \end{aligned} \tag{1}$$

where $t(u,v)$ is the mean luminance of the observed portion of an image in $\mathrm{cd/m^2}$, v is the velocity of the texture movement on the screen in $^{\mathrm{deg}}/_{\mathrm{s}}$, and $\boldsymbol{D}$ is a display configuration parametrized as:

$$\boldsymbol{D} = \{\widehat{\omega}, p, \widehat{\rho}, r_\mathrm{x}, r_\mathrm{y}\}, \tag{2}$$

where $\widehat{\omega}$ is the temporal Nyquist frequency of the display in Hz (half of display's refresh rate), p is the persistence (or duty-ratio) of the display ($0 < p \leq 1$), $\widehat{\rho}$ is the spatial Nyquist frequency of the display in cycles-per-degree (cpd) (half of display's spatial resolution in pixels-per-degree (ppd)), and $r_\mathrm{x}, r_\mathrm{y} \in \mathbb{Q}^+ \,|\, (0 < r_\mathrm{x}, r_\mathrm{y} \leq 1)$ are the shading resolution in x and y direction, respectively. The unit of Q is *Just-noticeable-difference* (JND).

Our model is partially inspired by Watson's analysis of capture and display systems [2013] and Denes et al.'s content-independent motion quality model [2020]. However, we focus on understanding the effect of texture content on motion quality to enhance and guide VRS in real-time graphics. For the rest of the paper, we will refer to our model as ***CaMoJAB***, a content-adaptive model of judder, aliasing, and blur.

4.1 Spatial distortion model

In a typical VRS rendering pipeline, a continuous texture goes through several stages before it is displayed (Figure 2). The texture is first quantized and stored as a mipmap. Next, during the texture lookup, a mipmap level is selected according to the shading resolution specified by VRS. The image is then displayed on the screen for a portion of the frame duration, which depends on the display persistence. If the object is in motion and is tracked by our gaze (smooth pursuit eye motion (SPEM)), it may introduce hold-type or eye motion blur. All the above stages add different kinds of spatial distortions to the signal which results in loss of quality. We will now analyze each of these steps in detail in the frequency domain (Figure 3).

Frequency domain analysis. Analyzing a visual signal in the frequency domain is a useful technique as it facilitates understanding sampling and filtering operations as well as allows for the use of various psychophysical models that are based on spatial frequency theory. Hence, we transform a given continuous 2D texture signal $t(u,v)$ moving with a velocity v in a horizontal direction, into its frequency domain representation $T(\tau_\mathrm{u}, \tau_\mathrm{v})$ using *Fourier transform*.

$$T(\tau_\mathrm{u}, \tau_\mathrm{v}) = \mathcal{F}\{t(u,v)\} \quad \left[^{\mathrm{cycles}}/_{\mathrm{texel}}\right], \tag{3}$$

where τ_u and τ_v are the spatial frequencies in cycles-per-texel (cpt) corresponding to u and v direction, respectively. Though we assume horizontal velocity in our model, we explain how it can be extended to arbitrary velocity in Section 5.

Variable rate shading. VRS executes the pixel shader once per multiple pixels and replicates the calculated color to all those pixels. The color typically involves a texture look-up operation on the *mipmap*

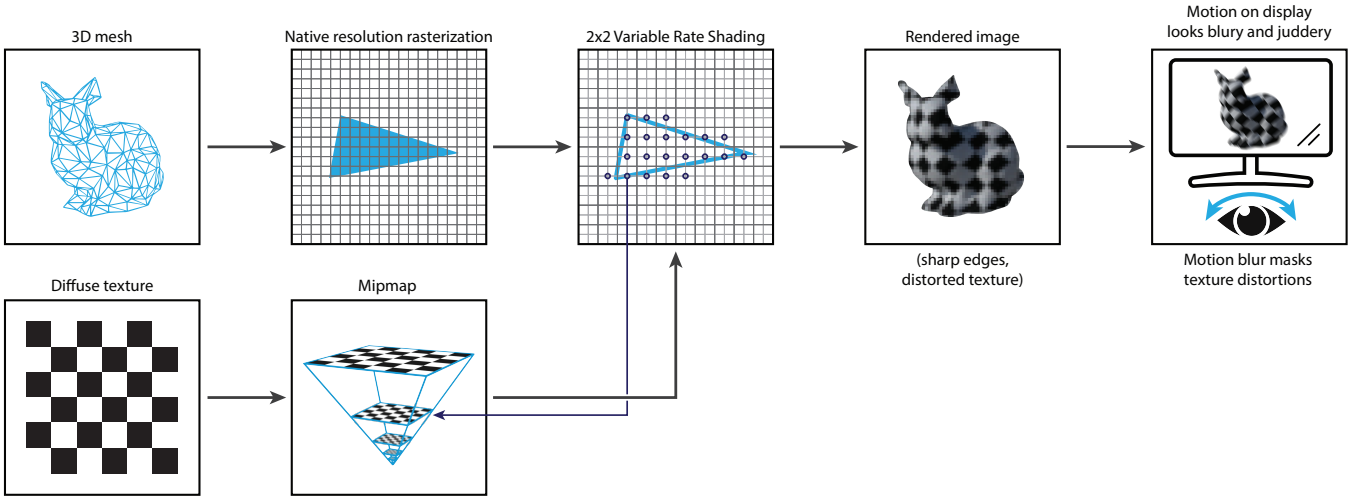


Fig. 2. Spatial and temporal distortions in a VRS pipeline. Rasterization happens at native resolution but pixel shader is called once per multiple pixels. Larger pixels size leads to selection of coarser mipmap levels yielding distorted texture in the rendered image (artifacts exaggerated for illustration). When the rendered image is displayed at discrete time intervals, the motion appears blurry and juddery. The motion blur may mask the texture distortions caused by VRS.

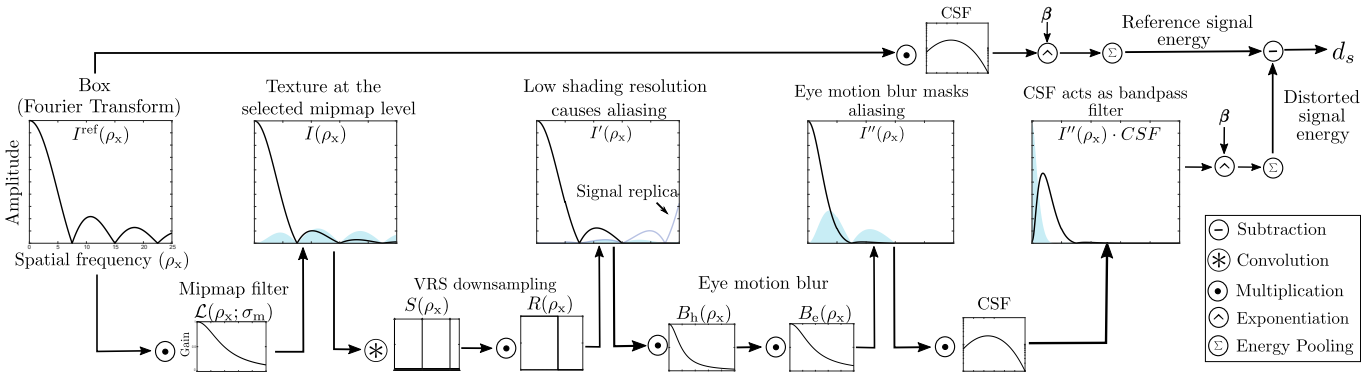


Fig. 3. Illustration of spatial distortions introduced when an image with a sharp edge (box) is moving on the screen. The black line shows the signal in the frequency domain and the blue shaded region denotes the distortion (difference between the current and previous signal). Picking a low shading resolution leads to a selection of coarser mipmap levels resulting in a loss of high frequencies. Low shading resolution also results in the formation of aliases that distorts the lower frequencies. When the rendered image moves on a screen, our eyes undergo SPEM to follow it. This adds further blurring of the signal and may mask aliasing. The above distortions can be mapped to a JND scale by calculating the difference of CSF-normalized visual energy of the distorted and the reference signal.

of the underlying albedo/diffuse texture. Mipmapping [Williams 1983] is the most popular method of antialiasing for textures. A mipmap is a pyramid structure created by low-pass filtering and downsampling to half the resolution of each fine level to create a coarser level. The common filters used for this purpose are box, Gaussian, Lanczos, and Kaiser [Akenine-Möller et al. 2019, Section 6.2.2]. In the following discussion, we assume a box filter as it is one of the most commonly used filters and yet is also known to be one of the worst possible filters because it unnecessarily blurs low frequencies while retaining some high frequencies that cause aliasing [Akenine-Möller et al. 2019]. For every shaded pixel, the area of the pixel is projected onto the texture. The projected area may include 1 or more texels. The level of detail to sample from is then selected in such a way that the ratio of pixel-to-texel area is close to 1:1. Most modern shading languages provide a function to calculate and query this value (such as *textureQueryLod()* in GLSL). Since VRS enlarges pixel area, it results in the selection of a coarser mipmap level. Let l be the mipmap level selected when rendering at full shading resolution such that $l = 0$ is the highest (finest) resolution level. The width of the box filter at lower shading resolution can be approximated as:

$$b_{\mathrm{m}} = \frac{2^l}{\min(r_{\mathrm{x}}, r_{\mathrm{y}})} \quad \left[\frac{\text{texel}}{\text{pixel}}\right]. \tag{4}$$

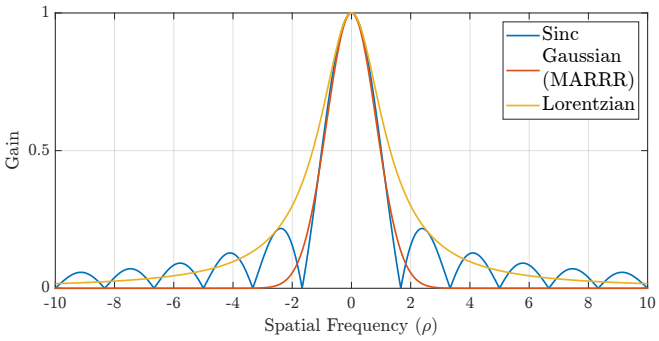


Fig. 4. Approximating sinc with monotonic functions.

The symbols used in this equation are defined in Eq. (2). By taking the minimum of the two shading resolutions, we assume isotropic filtering, but it is trivial to extend it to support anisotropic filtering. We will use upper case symbols to denote functions in the Fourier domain. The Fourier transform of our box filter gives:

$$B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m}) = \mathrm{sinc}(b_\mathrm{m}\tau_\mathrm{u}) \cdot \mathrm{sinc}(b_\mathrm{m}\tau_\mathrm{v})\,. \tag{5}$$

However, using a sinc(·) function introduces non-monotonicity in the model predictions making it unsuitable for optimisation problems. To mitigate this, we approximate sinc(·) with a Lorentzian function $\mathcal{L}(\cdot)$ of half-width σ_m:

$$\mathcal{L}(\tau; \sigma_\mathrm{m}) = \frac{\sigma_\mathrm{m}^2}{\sigma_\mathrm{m}^2 + 4\tau^2}\,, \tag{6}$$

$$\sigma_\mathrm{m} = \frac{\pi}{2b_\mathrm{m}}\,. \tag{7}$$

We found Lorentzian function to be a good approximation of sinc(·) as it preserves the high frequencies similar to sinc(·), unlike the Gaussian approximation used by [Denes et al. 2020] (refer to Figure 4).

The texture at a mipmap level l is therefore affected by a low-pass filter as follows:

$$T'(\tau_\mathrm{u}, \tau_\mathrm{v}) = T(\tau_\mathrm{u}, \tau_\mathrm{v}) \cdot B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m})\,, \tag{8}$$

$$B_\mathrm{m}(\tau_\mathrm{u}, \tau_\mathrm{v}; b_\mathrm{m}) \approx \mathcal{L}(\tau_\mathrm{u}; \sigma_\mathrm{m}) \cdot \mathcal{L}(\tau_\mathrm{v}; \sigma_\mathrm{m})\,. \tag{9}$$

Since we will now operate in screen-space, we map the mipmapped texture $T'(\tau_\mathrm{u}, \tau_\mathrm{v})$ to a screen space image $I\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right)$:

$$I\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right) = T'(\tau_\mathrm{u}\, 2\,\widehat{\rho}\, b_\mathrm{m}\,, \tau_\mathrm{v}\, 2\,\widehat{\rho}\, b_\mathrm{m}) \quad [\mathrm{cycles/deg}]\,, \tag{10}$$

where ρ_x and ρ_y are spatial frequencies in cpd.

VRS repeats the above texture look-up operation for all the "coarse" pixels (large VRS pixels). This is same as downsampling the filtered signal $I\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right)$ by convolving it with a sampling function $S(\rho_\mathrm{x}, \rho_\mathrm{y})$. The sampling function is a sequence of impulses at an interval of $2\,\widehat{\rho}\, r_\mathrm{x}$ and $2\,\widehat{\rho}\, r_\mathrm{y}$ in x and y direction, respectively. It can be represented using a Dirac comb function $\mathrm{Ш}(\cdot)$:

$$S(\rho_\mathrm{x}, \rho_\mathrm{y}) = \mathrm{Ш}\left(\frac{\rho_\mathrm{x}}{2\,\widehat{\rho}\, r_\mathrm{x}}\right) \mathrm{Ш}\left(\frac{r_\mathrm{y}}{2\,\widehat{\rho}\, r_\mathrm{y}}\right)\,. \tag{11}$$

In our implementation of $S(\rho_\mathrm{x}, \rho_\mathrm{y})$, we only considered the first two impulses to limit the discontinuities introduced by the function.

A display cannot render frequencies higher than its spatial Nyquist frequency. Hence, we can clip the high frequencies by multiplying the resultant signal with a rectangular function $R(\cdot)$:

$$I'(\rho_\mathrm{x}, \rho_\mathrm{y}) = \left(I\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right) * S\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right)\right) \cdot R\left(\rho_\mathrm{x}, \rho_\mathrm{y}\right)\,, \tag{12}$$

where the rectangular function is defined as:

$$R(\rho_\mathrm{x}, \rho_\mathrm{y}) = \Pi\left(\frac{\rho_\mathrm{x}}{2\,\widehat{\rho}\, r_\mathrm{x}}\right) \Pi\left(\frac{\rho_\mathrm{y}}{2\,\widehat{\rho}\, r_\mathrm{y}}\right)\,, \tag{13}$$

and $\Pi(\cdot)$ is a rectangular function[2].

We illustrate the above process in Figure 3 which analyses a moving box in the frequency domain. VRS picks coarse mipmap levels that act as a low-pass filter resulting in loss of energy. The filtered signal is then downsampled through a convolution with $S(\rho_\mathrm{x}, \rho_\mathrm{y})$ that results in the creation of replicas (*aliases*) of the spectrum at multiples of the sampling interval. It is followed by multiplication with $R(\rho_\mathrm{x}, \rho_\mathrm{y})$ which results in downsampling blur. If the bandwidth of the spectrum I is larger than display's effective Nyquist frequency ($\widehat{\rho}\, r_\mathrm{x}, \widehat{\rho}\, r_\mathrm{y}$), higher frequencies *fold* onto lower frequencies and cause spatial aliasing.

Eye motion blur. When the rendered image moves on the screen, our eyes undergo SPEM to track it. The motion of our eyes is continuous, however, an LCD presents the image at discrete moments in time (frames). This leads to smearing of the image on the retina and is known as *hold-type blur*. The hold-type blur can be reduced by reducing the time for which the frame is displayed (persistence). We follow the same practice as Denes et al. and approximate hold-type blur with a box filter of width:

$$b_\mathrm{h} = \frac{v\,p}{2\,\widehat{\omega}} \quad [\mathrm{deg}]\,, \tag{14}$$

where the parameters are the same in Eq. (2). Denes et al. showed that when our eye undergoes SPEM, its tracking is imperfect. Our eyes often over- or under-estimate the object location which leads to additional blurring. The degree of blurring is proportional to the object velocity. It can also be modeled with a box filter of width:

$$b_\mathrm{e} = (c_\mathrm{a}\, v + c_\mathrm{b}) \cdot p \quad [\mathrm{deg}]\,, \tag{15}$$

where $c_\mathrm{a} = 0.001648$ and $c_\mathrm{b} = 0.079818$ are the coefficients reported in their work.

The blur due to the hold-type blur and imperfect eye motion is modelled the same way as the loss of resolution due to VRS and mipmapping, using Lorentzian functions in the frequency domain (approximations of sinc filter):

$$B_\mathrm{h}(\rho_\mathrm{x}, \rho_\mathrm{y}) = \mathcal{L}(\rho_\mathrm{x}; \sigma_\mathrm{h}), \tag{16}$$

$$B_\mathrm{e}(\rho_\mathrm{x}, \rho_\mathrm{y}) = \mathcal{L}(\rho_\mathrm{x}; \sigma_\mathrm{e}), \tag{17}$$

where $\sigma_\mathrm{h} = \pi/2b_\mathrm{h}$ and $\sigma_\mathrm{e} = \pi/2b_\mathrm{e}$.

Note that the blurring due to eye motion is different from the blurring introduced in the previous stages (Eq. (9)). It is directional in nature and only happens parallel to object motion, unlike blurring due to mipmapping and downsampling which is omnidirectional.

The final distorted signal reaching our retina can be calculated as

$$I''(\rho_\mathrm{x}, \rho_\mathrm{y}) = I'(\rho_\mathrm{x}, \rho_\mathrm{y}) \cdot B_h(\rho_\mathrm{x}, \rho_\mathrm{y}) \cdot B_e(\rho_\mathrm{x}, \rho_\mathrm{y})\,. \tag{18}$$

[2]The rectangular function is equal to 1 in $(-0.5, 0.5)$, 0.5 at -0.5 and 0.5 and 0 otherwise.

Contrast sensitivity and energy pooling. To find the sensitivity of HVS to various distortions, we can modulate the signal with the spatio-temporal contrast sensitivity function (CSF) $S(\rho, \omega, L_a, r)$. For a given spatial frequency ρ, temporal frequency ω, adaptation luminance L_a, and stimulus radius r, the CSF specifies sensitivity, i.e., the reciprocal of minimum contrast threshold visible to an average observer. For our model, we use the CSF defined by Mantiuk et al. [2021] because it accounts for all three required dimensions together and is fitted to data up to 10 000 cd/m^2.

Finally, we can compute the overall visual energy E_s of the distorted signal by modulating it with the CSF and integrating over the entire spectrum:

$$E_{\mathrm{s}}(I'') = \int_0^{\widehat{\rho}} \int_0^{\widehat{\rho}} \left| I''(\rho_{\mathrm{x}}, \rho_{\mathrm{y}})\, S\left(\sqrt{\rho_{\mathrm{x}}^2 + \rho_{\mathrm{y}}^2}, \omega, L_a, r\right)\right|^{\beta} d\rho_{\mathrm{x}} d\rho_{\mathrm{y}}, \quad (19)$$

where ω is the temporal frequency, assumed to be 0, β is the power parameter of the model and is fitted to the psychophysical data in Section 4.4. We set the temporal frequency ω to 0 as we consider the stimulus that is stabilized on the retina and includes blur due to motion. We set the stimulus radius r to $\frac{1}{5}$ (the reciprocal of peak frequency). The adaptation luminance L_a is approximated as the mean luminance of the considered image part (VRS block). The CSF-normalised energy E_s can be linearly mapped to a JND scale as shown by [Denes et al. 2020].

Given a reference signal with no distortion

$$I^{\mathrm{ref}}(\rho_{\mathrm{x}}, \rho_{\mathrm{y}}) = T(2\widehat{\rho}\,\tau_{\mathrm{u}}, 2\widehat{\rho}\,\tau_{\mathrm{v}}) \quad [\mathrm{cycles/deg}], \quad (20)$$

we define the total spatial distortion (d_{t} in Eq. (1)) as the visual energy difference[3] between the reference signal and the distorted signal:

$$d_{\mathrm{s}} = E_{\mathrm{s}}(I^{\mathrm{ref}}) - E_{\mathrm{s}}(I''). \quad (21)$$

4.2 Temporal distortion model

When a continuous motion signal is sampled and displayed at discrete time intervals, the motion appears to be *jerky* or *juddery*. As discussed in Section 2.2, the amount of judder is known to increase with velocity and decrease with increasing refresh rate. This phenomenon can be explained by analyzing the signal in the frequency domain [Watson 2013].

We model the temporal judder (non-smooth motion) similarly as in [Denes et al. 2020] but in a content-adaptive manner. Temporal sampling creates replicas (aliases) of the signal in the frequency domain, each separated by the refresh rate of the display, $2\,\widehat{\omega}$ (see Figure 5). If the temporal bandwidth of the signal exceeds the Nyquist frequency of the display ($\widehat{\omega}$), the replica will overlap with the original signal and distort it. The temporal frequency of the first replica is $2\,\widehat{\omega}$ and the spatial frequency can be calculated as:

$$\rho_{\mathrm{A}} = \frac{2\widehat{\omega}}{v} \quad \left[\frac{\mathrm{cycles}}{\mathrm{deg}}\right]. \quad (22)$$

We can safely ignore the y spatial frequency as we had assumed the velocity in y to be 0 and so no temporal distortion. The total temporal distortion (d_{t} in Eq. (1)) can then be approximated as

[3]Different from Watson and Ahumada's [2011] visible contrast energy (ViCE). We found the above formulation to be more robust to numerical inaccuracies.

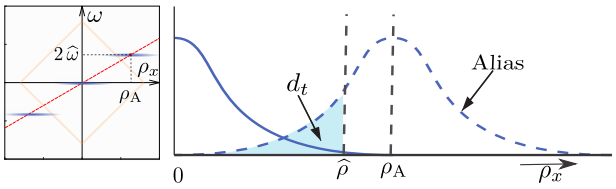


Fig. 5. Temporal distortion of the signal due to low refresh rate.

Table 3. Range of conditions tested in each block. The same 4 textures (Checkerboard, Noise, Grass, and Gradient) processed to the correct luminance and contrast were used for all blocks.

	Refresh Rate (Hz)	Persistence	Velocity (deg/s)	Resolution (ppd)	Luminance (cd/m^2)	Michelson Contrast
Block-PC	144	1	0, 10, 30	50	150	.7, .8, .9
Block-Mobile	60	1	3, 10, 20	90	75	.7, .8, .9
Block-VR	144	0.05	10, 45, 75	13	2.5	.7, .8, .9

the CSF-normalised visual energy in the region of the replicated spectrum that is less than the spatial Nyquist frequency of the display ($\widehat{\rho}$):

$$d_{\mathrm{t}} = \int_0^{\widehat{\rho}} \int_0^{\widehat{\rho}} \left| I'(\rho_{\mathrm{x}} + \rho_{\mathrm{A}}, \rho_{\mathrm{y}})\, S\left(\sqrt{\rho_{\mathrm{x}}^2 + \rho_{\mathrm{y}}^2}, 2\widehat{\omega}, L_{\mathrm{a}}, r\right)\right|^{\beta} d\rho_{\mathrm{x}}\, d\rho_{\mathrm{y}}\,, \quad (23)$$

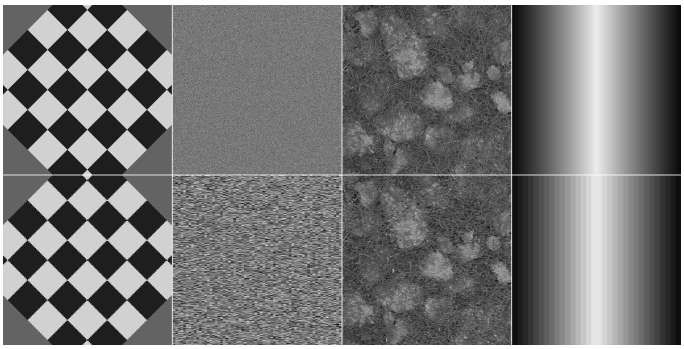
Similar to Eq. (19), we approximate adaptation luminance L_a as the mean luminance of the considered image part (VRS block) and set the stimulus radius r to $\frac{1}{5}$.

4.3 Experiment 1: VRS Motion Quality Measurement

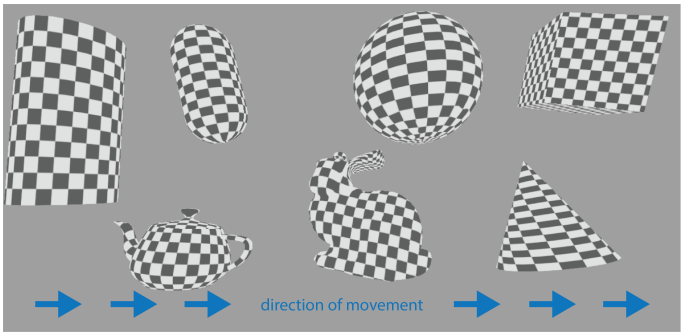
We found existing datasets to be insufficient to calibrate and test our metric. This is because the metric accounts for new factors, such as shading rate and persistence, which have not been well studied before. For that reason, we conducted a motion quality experiment, which was designed to measure the effect of those less explored factors. Here, we give a high-level overview of the experiment, while the detailed description can be found in Appendix A.

The experiment involved the comparison of animations shown on two 165 Hz adaptive-sync monitors, both running at the same refresh rate. The animations consisted of several 3D objects (shown in Figure 6(b)), all textured with one of the gray-scale textures shown in Figure 6(a) and moving horizontally with a constant speed. Both animations differed only in the shading rate or the texture used. The participants were asked to select the animation of higher quality.

Results. The pairwise comparison results from this experiment were mapped to a Just-Noticeable-Difference (JND) scale based on Thurstone Case V assumptions [Perez-Ortiz and Mantiuk 2017]. A quality difference of 1 JND means that 75% of the population will pick one condition over another. Since the original formulation of Thurstone Case V assumption does not allow for the computation of confidence intervals, bootstrapping was used to compute them. The results are reported in Figure 7a. The shading rate of 1×2 was assigned as the 0 JND condition and other data points are presented



(a) Textures used in the experiment (top row) and their counterparts rendered using a lower shading rate (bottom row). The VRS was simulated and exaggerated using nearest neighbor downsampling.



(b) Horizontally moving and slowly rotating 3D models, rendered with the checkerboard texture, that were shown in the experiment.

Fig. 6. Textures and objects used in the experiment.

relative to it. The shapes of the plots are consistent with previous studies with quality impairment (ΔQ) due to reduced shading resolution decreasing with increasing velocity. Though this effect is not as prominent in VR. This can be explained by hold-type blur increasing with velocity and masking the spatio-temporal aliasing caused by reduced shading rate. Since VR has minimal hold-type blur because of its low-persistence, the VRS artifacts continue to be visible even at high velocities. Another important observation is that quality degrades non-linearly with shading resolution and the slope of this curve varies widely between different textures; with artifacts in gradient rarely visible and artifacts in noise almost always visible. From these results, we confirm that the motion quality depends on a wide range of display and content related factors.

4.4 Model calibration

To determine the free parameter β and the relative weights of the spatial and temporal distortions w_s and w_t, we use the psychophysical data from our Experiment 1 (Section 4.3) in addition to Experiment 1 and 3 from [Denes et al. 2020]. The experiment by Denes et al. measured supra-threshold motion quality through moving checkerboards at different refresh rates and velocities on a full persistence LCD and no-reference pairwise comparisons. The two datasets use the same JND scale and together include a wide range of display settings and all three artifacts, i.e., motion blur, downsampling artifacts, and judder (refer to Table 1 for the exact range of factors tested). The values of the parameters are determined by minimizing the root-mean-squared error between the model predictions and the results of the experiments. The fitted parameter values (RMSE = 0.53) are reported in Table 4 and the results are plotted in Figure 7a and 7b. CaMoJAB correctly predicts the reduction in quality differences with increasing velocity and successfully captures the relationship between different textures across all three configurations. An ablation study analyzing the contribution of each of the CaMoJAB's component can be found in the supplementary.

Table 4. Model parameters

w_s	w_t	β
3.69	2.9	0.36

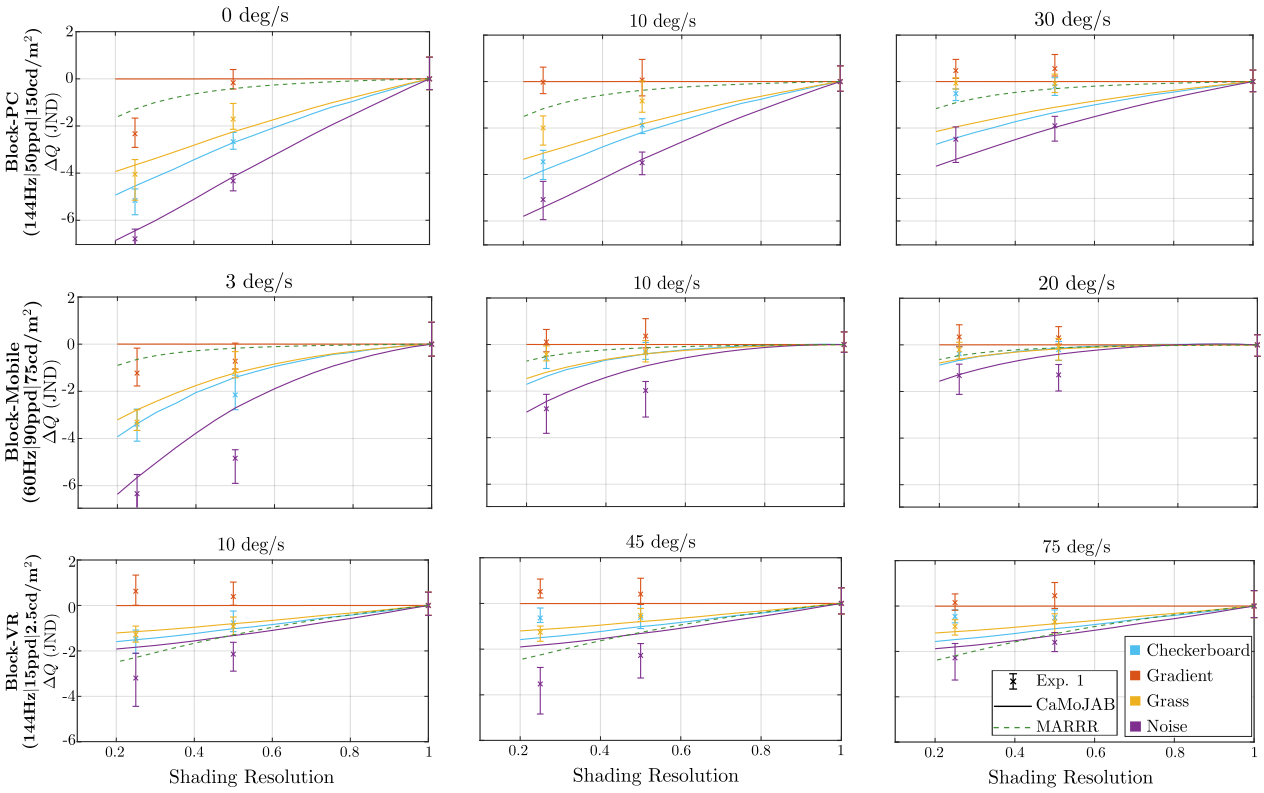
4.5 CaMoJAB Results and comparison

We provide a visualization of CaMoJAB predictions for the moving grass texture at different velocities, refresh rates, and shading resolutions in Figure 8. In the first plot, it predicts that as the velocity (in $^{deg}/_{s}$) increases, the difference in quality between high and low shading resolution decreases. This is consistent with our observations in Experiment 1 (Section 4.3). In the second plot, it predicts that at lower velocities, observers prefer higher shading resolution over refresh rate and the trend reverses as the velocity increases. This is consistent with the findings of [Denes et al. 2020]. These model predictions hold high practical significance as they can be used to drive real-time rendering on performance adaptive hardware such as variable refresh rate (VRR) displays or VRR compatible cloud gaming [Colenbrander 2021], and VRS GPUs. We describe and implement one such strategy in a complex motion setting in Section 5. Finally, our model also provides an opportunity to study the interplay between luminance and persistence (Figure 9) for rendering on the upcoming variable persistence-variable refresh rate displays [Hekstra et al. 2008; Verbeure et al. 2017].

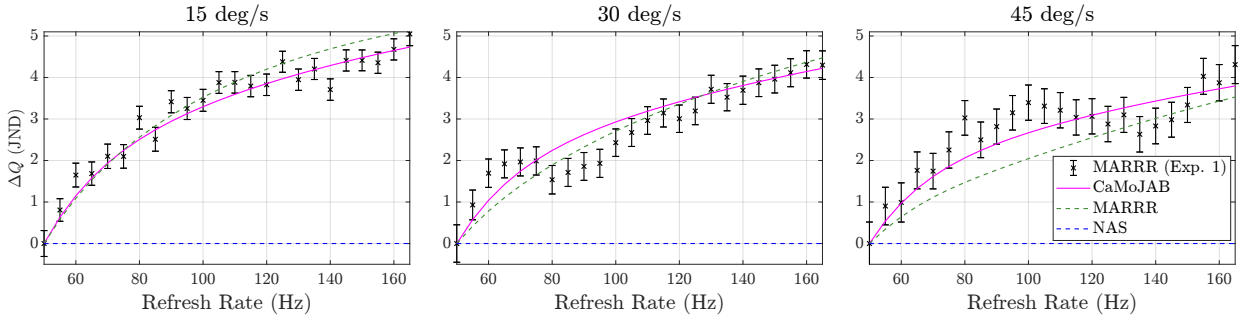
For a comparative study, we picked two state-of-the-art models [Denes et al. 2020] and [Yang et al. 2019]. It should be noted that neither of these models (or any other model to our knowledge) is designed to predict our kind of data. [Denes et al. 2020] is a perceptually based model derived through extensive psychophysical experiments, however, it is content-independent. [Yang et al. 2019] though content dependent, does not take display and viewing conditions into consideration. Furthermore, it is designed for a specific task to give quality scores for only full, half, and quarter shading resolution and not a continuous scale. For a fair comparison, both models were linearly fitted to our data. As can be seen from the quantitative comparison in Table 5, our model predicts the trends significantly better than the other two models across all datasets. For a more extensive comparison, please refer to the supplementary.

4.6 CaMoJAB Validation

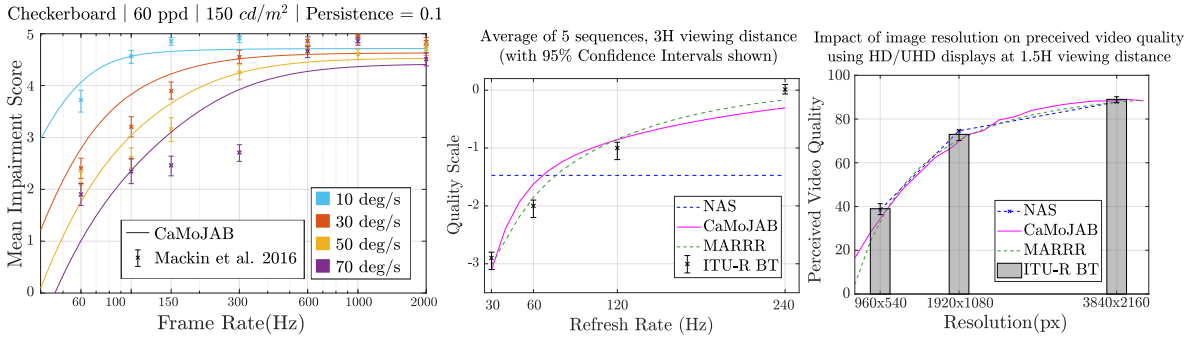
To further validate and verify the generalizability of our model, we test our model's performance on motion quality datasets from



(a) Experiment 1 results and CaMoJAB predictions scaled to JND units.



(b) CaMoJAB predictions on [Denes et al. 2020]'s Experiment 1.



(c) CaMoJAB's validation on [Mackin et al. 2016] and ITU-R [Series 2020]. It generalizes well to complex stimuli such as natural videos and real scenes.

Fig. 7. Experiment 1 results and CaMoJAB predictions. CaMoJAB was calibrated together on all the results reported in (a) and (b) and is capable of capturing the correct trends. MARRR being a content-independent model, is unable to capture the difference in quality due to texture in (a). NAS does not account for display factors and thus fails to predict the change in quality due to refresh rate in (b).

Table 5. MARRR, NAS, and CaMoJAB (Ours) performance on motion quality datasets. All models were linearly fitted to each dataset. Numbers denote the RMSE values. We report RMSE corresponding to a moving checkerboard for the ITU dataset as they did not report the stimulus used.

	Block-PC	Block-Mobile	Block-VR	MARRR Exp. 1	MARRR Exp. 3	Mackin et al. [2016]	ITU-R FPS [Series 2020]	ITU-R PPD [Series 2020]
CaMoJAB (ours)	**0.74**	**0.57**	**0.53**	**0.38**	**0.03**	**0.42**	0.28	0.69
MARRR	1.82	1.55	1.14	0.55	0.04	0.55	**0.15**	**0.25**
NAS	1.41	1.1	1.23	3.12	0.21	1.04	1.09	1.19

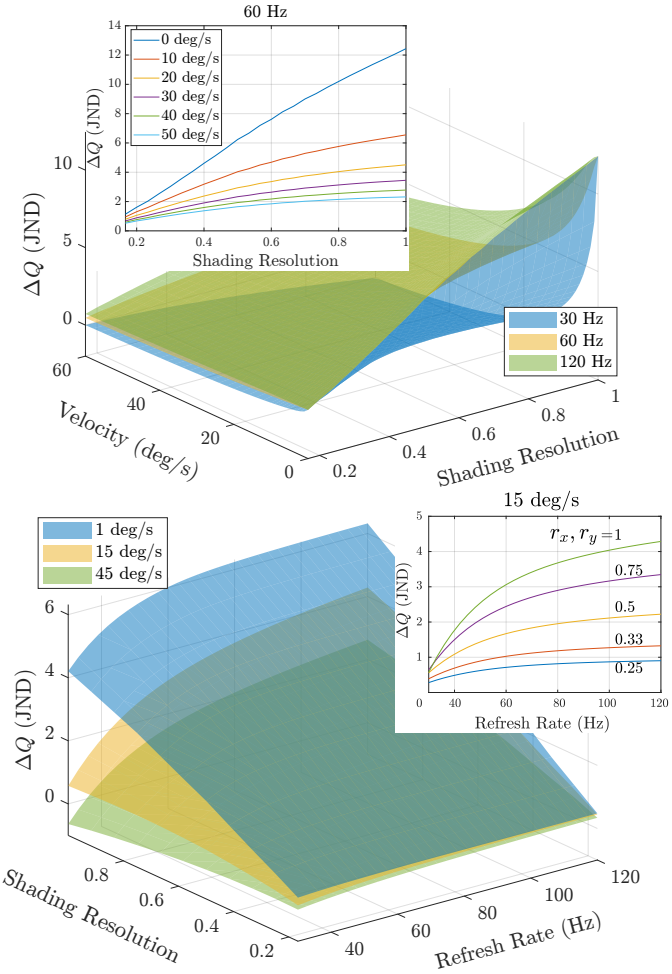


Fig. 8. CaMoJAB predictions for a moving grass texture on a 150cd/m^2 and full persistence 60 ppd display. Although we plot shading resolution as a continuous variable, in practice, only a small set of resolutions are available.

[Mackin et al. 2016] and ITU-R recommendations [Series 2020]. The experiment by Mackin et al. used mean opinion score methodology to study motion quality of real-continuous motion of rotating lines on a low persistence display (simulated via strobing lights on a printed paper). The experiments in ITU-R [Series 2020] use natural videos as stimuli and their results form the basis of presently in-force ITU recommendations.

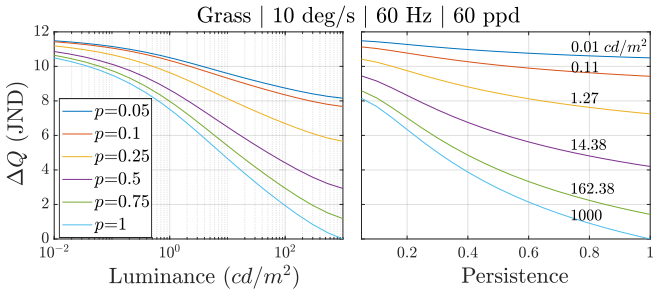


Fig. 9. Luminance-persistence trade-off for a 60 Hz animation on a 60 ppd display. The motion artifacts in such animation are more noticeable at high luminance and for longer display persistence.

We linearly scaled our model to fit their data and bring the predictions to the same perceptual scale (Figure 7c). We report the goodness of fit (RMSE) for a moving checkerboard in Table 5. The results show that our model also generalizes well to more complex stimuli.

5 ADAPTIVE LOCAL SHADING AND REFRESH RATE

Modern graphics engines require dynamic quality control that would adapt rendering parameters (resolution, refresh rate, level of detail, and others) to the available resources (GPU frame-budget, power). This becomes especially important for mobile devices with high-resolution and refresh-rate displays, but also in time-sharing cloud rendering, used for streaming of games. Current solutions used in popular game engines typically involve dynamic resolution scaling of the entire frame to meet these constraints [Unity 2021; Unreal 2021]. This is a sub-optimal approach, as better quality can be obtained by (a) adaptively selecting the best combination of resolution and refresh-rate [Denes et al. 2020] and (b) adjusting the resolution locally, using VRS. In this section we describe our Adaptive Local Shading and Refresh Rate (**ALSaRR**) algorithm that uses our motion quality metric to determine the optimal distribution of shading rate and refresh rate under a given bandwidth constraint.

An overview of our ALSaRR can be found in Figure 10. It takes as input the allowed bandwidth B in pixels per second and several auxiliary buffers rendered at the resolution of the VRS map: motion vectors, texture IDs, mipmap levels and luminance. The pooled motion vectors and the bandwidth B are used to determine the best frame rate for the current frame. Then, the maps are fed to the CaMoJAB to compute for each VRS block the ratios of quality to bandwidth for all possible shading rates. The ratios are then used by a greedy knapsack solver to find the distribution of shading rates that

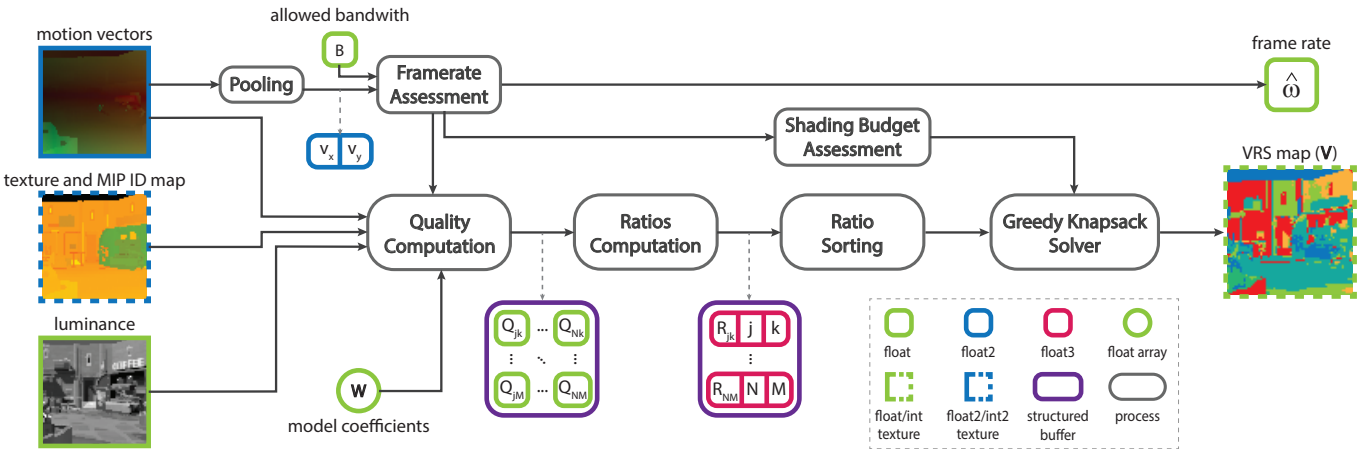


Fig. 10. The processing diagram for Adaptive Local Shading and Refresh Rate (ALSaRR). The method uses motion vectors, luminance, and precomputed polynomial coefficients to predict the optimal refresh rate and distribution of the shading rates given allowed bandwidth.

maximize the quality at the given budget. In the following sections, we describe each step in detail and how they can be implemented in a real-time game engine.

5.1 Auxiliary buffers

Our quality metric requires the knowledge of texture, the mapping of texels to fragments, luminance, and velocity of the pixels rendered on the screen. For that purpose, we render into a G-buffer: texture IDs, MIP map level, motion vectors, and luminance at the resolution of the VRS map (width/16 × height/16 pixels). As rendering luminance would require additional shading, which we want to avoid, we use the previously rendered frame (without reprojection) to approximate luminance. The previous frame is first subsampled, then converted to grayscale luma, from luma to linear absolute luminance values, and scaled according to the display peak luminance.

5.2 Optimal refresh rate

Adaptive-sync displays allows us to chose arbitrary refresh rates, which let us either improve smoothness of motion or spatial resolution, depending on the velocity of motion. Similar to MARRR, we express this as an optimization problem:

$$\begin{aligned} &\arg\max_{\widehat{\omega}} q\left(I, v_{\text{frame}}, \widehat{\omega}, p, \widehat{\rho}, r_{\text{x}}, r_{\text{y}}\right) \\ &\text{s.t.} \quad 2\,\widehat{\omega}\,R_{\text{x}}\,r_{\text{x}}\,R_{\text{y}}\,r_{\text{y}} = B \;\wedge r_{\text{x}} = r_{\text{y}} \wedge\; 2\,\widehat{\omega} \in \mathbb{Z}^{+} \;\wedge\; 2\,\widehat{\omega} \le \omega_{\max}, \end{aligned} \tag{24}$$

where q is the quality function defined in Eq. (1), R_x and R_y are the horizontal and vertical display resolution in pixels, $\omega_{\max}$ is the maximum refresh rate of the display and B is the fixed budget in pixels-per-second. As the refresh rate is global for the frame, we take the entire screen-space image (I) as a proxy for the image being tracked and pool the motion buffer to compute the average velocity of the frame (v_{frame}) in deg/s. The rest of the parameters are the same as in Eq. (2).

Once we have the optimal refresh rate $2\widehat{\omega}$, we can calculate the maximum allowable shading budget of the given frame as

$$b_{\text{frame}} = \frac{B}{2\widehat{\omega}} \quad [pixels]. \tag{25}$$

Since solving the optimization for every frame is too expensive for real-time rendering, we precompute the best combinations using the first frame and store them as a 1D LUT of velocity vs. refresh rate (and resolution).

5.3 Optimal shading rates

In this section, we describe how to obtain an optimal VRS map under a limited shading budget b_{frame}.

Problem formulation. A screen-space image can be divided into N 16 × 16 pixel VRS tiles ($\boldsymbol{\mathcal{T}}_j$, $j = 1, \ldots, N$). Each tile $\boldsymbol{\mathcal{T}}_j$ has a bandwidth ($\boldsymbol{B}_{jk}$) and a quality ($\boldsymbol{Q}_{jk}$) corresponding to k^{th} shading rate from the set of all available shading rates $\boldsymbol{\mathcal{R}}$.

$$\boldsymbol{\mathcal{R}} = \{4 \times 4, 4 \times 2, \ldots, 1 \times 1\} \tag{26}$$

such that for $\boldsymbol{\mathcal{R}}_k = m \times n$. The bandwidth can be calculated as

$$\boldsymbol{B}_{jk} = \frac{256}{m \cdot n} \quad [\text{pixels}]. \tag{27}$$

Since our motion quality model assumes unidirectional motion, we calculate the quality in x and y direction separately using the quality function from Eq. (1), so that

$$\begin{aligned} q_x(\boldsymbol{\mathcal{T}}_j, \boldsymbol{D}, \boldsymbol{\mathcal{R}}_k) &= q\left(\boldsymbol{\mathcal{T}}_j, v_j^{\text{x}}, \widehat{\omega}, p, \widehat{\rho}, \frac{1}{m}, \frac{1}{n}\right) \quad [\text{JND}], \\ q_y(\boldsymbol{\mathcal{T}}_j, \boldsymbol{D}, \boldsymbol{\mathcal{R}}_k) &= q\left(\boldsymbol{\mathcal{T}}_j^{\top}, v_j^{\text{y}}, \widehat{\omega}, p, \widehat{\rho}, \frac{1}{n}, \frac{1}{m}\right) \quad [\text{JND}], \end{aligned} \tag{28}$$

where v_j^{x} and v_j^{y} are the minimum velocity in the tile j along x or y (the most conservative assumption). The rest of the parameters are the same as in Eq. (2). We define the total quality of the tile as the mean of q_x and q_y

$$\boldsymbol{Q}_{jk} = \left(\frac{q_x(\cdot) + q_y(\cdot)}{2}\right) \quad [\text{JND}]. \tag{29}$$

Finally, given the maximum allowable bandwidth per frame b_{frame}, we need to select the shading rate k for each tile $\mathcal{T}_j$ to find

$$\arg\max_{\boldsymbol{B}_{jk}} \sum_{j=1}^{N} \boldsymbol{Q}_{jk} \quad \text{subject to} \quad \sum_{j=1}^{N} \boldsymbol{B}_{jk} < b_{\text{frame}} \tag{30}$$

Greedy solution. The optimal solution of the above knapsack problem can be found using dynamic programming but the algorithm has the time complexity of $\Theta(N \cdot b_{\text{frame}} \cdot |\mathcal{R}|)$ and the space complexity of $\Theta(N \cdot b_{\text{frame}})$, making it unsuitable for real-time rendering (alone $N = 8100$ for the FullHD resolution). Instead, we solve the problem using an approximated greedy solver [Kellerer et al. 2004, Section 2.1], which has the time complexity of $\Theta(N \cdot |\mathcal{R}|)$ and the space complexity of $\Theta(N \cdot |\mathcal{R}|)$. The algorithm finds near-optimal allocation of shading rates based on the quality predictions. Its pseudocode can be found in Algorithm 1.

Let $\boldsymbol{V}$ be a vector of N elements storing the current shading rate for each tile j in $\boldsymbol{V}_j$. We initialize $\boldsymbol{V}$ with the index of lowest bandwidth shading rate, $\boldsymbol{V}_j = 1 \; \forall \; j \in [1, N]$, and subtract the cost of rendering at that rate from the current frame bandwidth budget, b. Next, we calculate all possible ratios of quality to bandwidth for each tile, $\boldsymbol{R}_{jk} = \boldsymbol{Q}_{jk}/\boldsymbol{B}_{jk}$, and sort them in non-increasing order. The first element in the sorted list will give the tile and shading rate that increases the quality at the least cost (bandwidth). Next, we greedily pick (j, k) pairs from the head of the sorted list and update the current frame budget, $b = b - \boldsymbol{B}_{jk} + \boldsymbol{B}_{j\boldsymbol{V}_j}$ and shading rate of j^{th} tile, $\boldsymbol{V}_j = k$. The steps are repeated until the remaining budget becomes 0.

Algorithm 1: Greedy solution for the modified knapsack problem.

```
Result: near-optimal shading rate of each tile
b ⟵ b_frame
for j = 1 to N ;                                  // For each tile
 do
    V_j ⟵ 1 ;                                     // Shading rate 4×4
    b ⟵ b - B_j1 ;                                // Update remaining bandwidth
    for k = 2 to |R| ;                            // For each shading rate
     do
        R_jk ⟵ Q_jk / B_jk ;                      // Calculate ratios
    end
end
sort(R) ;                                         // From the highest to the lowest
while b > 0 do
    (j,k) ⟵ R.pop() ;                             // Pick the highest ratio
    b ⟵ b - B_jk + B_jV_j ;                       // Update the bandwidth
    V_j ⟵ k ;                                     // Update the shading rate
end
```

To ensure that the near-optimal solution is sufficient, we compared the results of the greedy algorithm with the accurate solution. Although the theoretical error in the estimated near-optimal solution can be up to 50% of the optimal objective function value, we found that in practice the greedy method has the expected error of less than 1% in our tested scene while having orders of magnitude lower time and space complexity. Details on this can be found in the Supplementary.

Real-time implementation. To reduce the performance overhead of our method, the quality predictions of CaMoJAB (Eq. (1)) are precomputed for every texture, mipmap level, and shading rate and stored as polynomials of the form:

$$\begin{aligned} q_{i,j,k}(v, f, L) = (W_0 + W_1 v + W_2 f + W_3 v^2 + W_4 v f + W_5 f^2) \\ \cdot (W_6 L^3 + W_7 L^2 + W_8 L + W_9), \end{aligned} \tag{31}$$

where i is the texture index, j is the MIP map level, k is the shading rate, v is velocity, f is the refresh rate of the current frame ($2\,\widehat{\omega}$), L is the logarithmic luminance of the tile, and $W_{0,\ldots,9}$ are the coefficients of the polynomial, stored separately for each (i, j, k) triplet.

Our real-time implementation follows Algorithm 1 and computes the ratios of quality to bandwidth for each tile and each possible shading rate. Those ratios are sorted in a non-increasing order (the best quality-to-shading rate first) using a bitonic sort, as this algorithm can utilize the parallelism of the GPU.

5.4 The velocity of eye motion without eye tracking

In a typical animation, multiple objects can move on the screen in different directions with different velocity and the eye could follow any of them. For that reason, MARRR method required an eye tracker to determine the actual velocity of the eye motion relative to the screen. However, because we control the shading rate locally, we can make a conservative assumption that the eyes could follow each of 16×16 tiles. This lets us use different velocity of eye motion for each tile instead of a global velocity per frame, as done in MARRR. Because the distortions are the most visible when the object is followed, we get the worst-case estimate of the degradation of quality due to the reduced shading rate. This gives us an important practical advantage over MARRR as eye tracking is still rare in most applications of real-time graphics.

5.5 Implementation details

To test our method on realistic scenes of sufficient complexity, we implemented a native plugin for Unity. The plugin modifies the default forward rendering pipeline by allowing for a custom VRS map. VRS is enabled only for the fragment shader stage and is disabled for the post-processing pass, as that would introduce reduced shading at the pixel level and shade fragments across the edges.

Performance. We measured the performance of ALSaRR on a GeForce RTX 2080Ti graphics card using built-in Unity profiler. Our implementation utilizes parallelism of GPU and requires, on average, 1.3 ms to process the frame of resolution 1920 × 1080 pixels. It also requires approximately additional 100 MBs of VRAM to store intermediate data. The processing time is independent of the rendered content and does not include preprocessing (fitting of the polynomials, Eq. (31)). We did not attempt to implement the method on a mobile GPU but we expect the efficient implementation on mobile architectures may require further performance optimizations,

such as computing VRS maps at lower resolution and upsampling them.

5.6 Experiment 2: ALSaRR for real-time rendering

In this section, we report on a pairwise-comparison preference experiment, which compared our ALSaRR method against constant-refresh-rate-resolution and MARRR. The experiment simulated video game scenarios, all implemented in Unity 3D and rendered with one of the compared methods.

Stimuli. We adapted 3 scenes from Unity Asset store (Figure 11). For every scene, we prepared 4 different camera motion scenarios. One of them was mimicking a first- or third-person game, allowing the participants to control the camera with the mouse and keyboard. The rest of them contained predefined motion paths, each one with a different velocity of camera motion. Each scene was rendered using one of the three compared methods: ALSaRR, constant-refresh-rate-resolution, and MARRR. For the constant-refresh-rate-resolution method, we used several combinations of shading and refresh rates, each with a fixed budget of 6.25% or 12.5% pixels of the full bandwidth (1920×1080 @ 120 Hz). We used a lower resolution than the native resolution of the display because we noted in the pilot experiments that participants have difficulty distinguishing between the conditions at higher resolutions (on a 27" monitor seen from 75 cm).

MARRR method originally required an eye tracker to determine the velocity of eye motion (relative to the screen). However, for closer comparison, we made the same simplifying assumption as we did for the selection of refresh rate in our method and estimated the eye motion velocity as the average velocity within the entire frame (pooled from the motion buffer).

The rendered scenes were shown on the same monitor as used in Experiment 1 (Section 4.3).

Procedure. The experiment consisted of pairwise comparisons between the three compared rendering methods. In each trial, one of the conditions was our method and the other condition was either a constant-resolution-refresh-rate or MARRR. The order of the scenes and the compared methods was randomized. The camera motion scenario was randomly selected for each trial as a full factorial design was infeasible within a reasonable time budget.

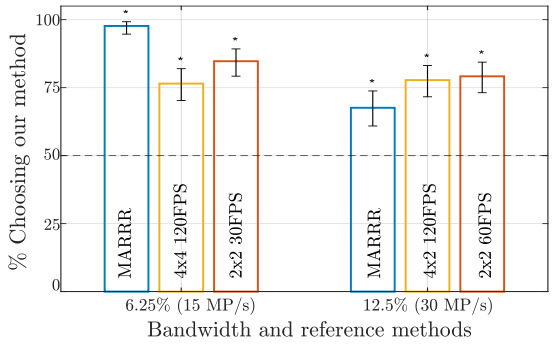
The participants could press the *space* bar to switch between two compared rendering methods. They had to choose the rendering method with overall better visual quality by pressing the *Enter* key while the preferred rendering mode was active. The question shown to the observers was: *"Which of the two methods has higher visual quality (i.e. smooth motion and sharp image)?"*. Each observer performed 108 comparisons (3 scenes × 6 budgets/rendering methods × 6 repetitions). To prevent fatigue, the experiment was split into 2 sessions lasting 30 minutes each.

Participants and procedure. 12 participants (aged 24-41) with normal or corrected-to-normal vision participated in the experiment. They were compensated for their time.

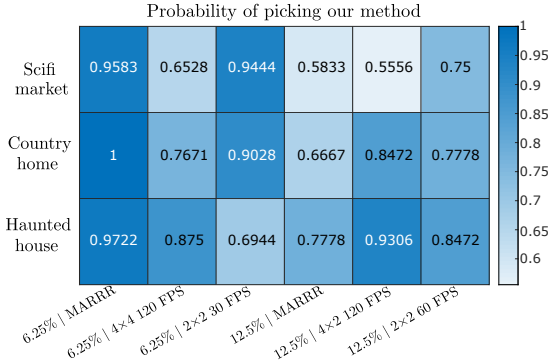
Fig. 11. Preview of the scenes selected for validation experiment. The *SciFi Market* scene[4] (top) and *John Lemon's Haunted Jaunt* scene[5] (bottom) have been released as learning materials by Unity. The *Sally's Country House* scene[6] (middle) created by *Gabro Media* has been acquired from Unity Asset Store.

Results. The results of the validation experiment, visualized in Figure 12, show a clear preference for our ALSaRR method, as compared to fixed resolution rendering and MARRR. The difference is significant across all tested conditions (binomial test with the null hypothesis of random guess). As expected, the improvement due to adaptive rendering decreases as the bandwidth increases because the differences between the conditions become less noticeable. MARRR performed better than fixed-frame-rate at a higher budget of 12.5%, but worse at 6.25%. It should be noted, however, that our fixed-rate conditions used reduced shading rate instead of bilinear upsampling used in [Denes et al. 2020]. This may indicate that reducing the shading rate may produce better visual results than upsampling from a lower resolution when the rendering budget is low. The lack of eye tracking could also negatively affect the performance of MARRR.

[4]https://assetstore.unity.com/packages/essentials/tutorial-projects/adventure-sample-game-76216
[5]https://learn.unity.com/project/john-lemon-s-haunted-jaunt-3d-beginner
[6]https://assetstore.unity.com/packages/3d/environments/urban/sally-s-country-home-33123



(a) Results of Experiment 2, aggregated across all scenes



(b) Results of Experiment 2 for each scene

Fig. 12. Results of the validation experiment showing percentage of participants picking our method over MARRR and constant-refresh-rate-resolution rendering. Error bars in (a) denote 95% confidence interval.

The per-scene results, shown in Figure 12(b), suggest the performance gain of our method was the smallest for the "Scifi market" scene. After closer inspection, we noted that the scene contained a large number of hard shadows, which resulted in aliased edges for our method. This is because our current implementation of ALSaRR relies on the information from textures, rather than rendered fragments and, therefore, ignores local shading (with the exception of mean luminance). This problem is difficult to mitigate in real-time rendering as the decision on shading rate must be made before local shading is computed.

6 LIMITATIONS AND DISCUSSION

One of the important advantages of our method over foveated rendering [Tursun et al. 2019] and MARRR is that it does not require eye tracking. However, to eliminate the dependency on eye tracking, we had to make a conservative assumption that the eye can follow movement in every VRS tile. If the gaze location was known to us, we could potentially further improve the predictions and therefore allocation of the rendering budget. Our method could be also combined with foveated rendering for further gains in quality.

Our CaMoJAB metric accounts for most relevant factors of motion quality, including texture content, on-screen velocities, luminance, effective resolution, and display persistence. However, it does not model contrast masking [Mantiuk et al. 2021], degradation of color,

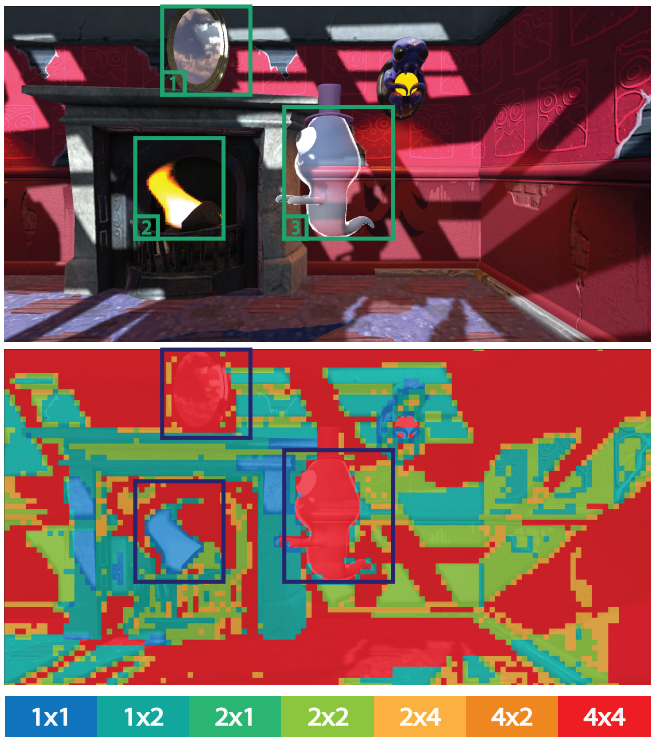


Fig. 13. Examples of fail cases: because ALSaRR relies on textures rather than shaded pixels, it assigns low shading rates to mirror reflections (1) and transparent ghost (2); high shading rate to fire particles (3).

foveation [Tursun et al. 2019] or saliency. Those limitations are partially dictated by the scarcity of available data, and partially by the real-time requirement of our application. The complexity of contrast masking would make the metric too costly for a real-time application, and foveation would require eye tracking. Also, CaMoJAB does not account for individual variations and instead models the average observer. Modeling individual variations would require much more measurements from each participant.

In comparison to *Nvidia Adaptive Shading* (NAS) [Yang et al. 2019], CaMoJAB is calibrated in physical units (pixels-per-degree, degrees-per-second, cd/m^2) and accounts for the display characteristics that affect motion quality perception, such as refresh rate, duty-cycle, and field-of-view. NAS may need to be recalibrated for a display of different size, brightness or pixel density than the one used by the authors. CaMoJAB has also been calibrated with several datasets, collected using psychophysical procedures, rather than tuned by the authors.

Practical considerations for ALSaRR. Our ALSaRR method can control real-time rendering with an acceptable overhead because we can pre-compute the quality predictions for all textures, mipmap levels, and shading rates. The main shortcoming of this approach is that our CaMoJAB quality metric is unaware of the shading in the final image, for example, when a tile contains a shadow boundary. A few such failure cases are demonstrated in Figure 13 where ALSaRR assigns non-optimal shading rates due to the lack of correct

frequency information. The examples shown in the figure include (1) reflections, (2) transparent objects and (3) procedural particle systems. Some game engines may not support rendering motion vectors for transparent materials further exacerbating the problem. It should be noted that ALSaRR uses diffuse textures as a proxy for object frequency information. Though this served as a good approximation for our tested scenes, it may not hold true for more complex materials. These issues can be addressed to an extent by precomputing the worst-case scenario shading and then using that for our precomputed quality functions. We could also use the approach from NAS [Yang et al. 2019] and directly analyze the frequency information by reprojecting the previous frame. This approach, however, relies on the assumption that the previous frame has been rendered with sufficient quality, and it also adds a substantial overhead of reading and processing full-resolution frames.

Our current implementation of ALSaRR operates on the rendering budget expressed in pixels per second while most applications are limited instead by the power of GPU cycles. This is not a fundamental limitation of our method, as it can work with the budget expressed in any units as long as there exists a model that can predict such units from the VRS rates.

7 CONCLUSIONS

There is a shift in real-time graphics from rendering with a fixed resolution and refresh rate to a more adaptive approach, in which we control spatio-temporal resolution in order to maximize the quality under a given rendering budget. VRS and adaptive refresh-rate monitors provide an opportunity for such a fine control, which becomes particularly important as the increasing resolution and refresh rate of displays places a high demand on the GPU, especially on mobile devices. Our ALSaRR method takes the advantage of the limitations of the visual system to reallocate rendering budget to the most vital part of the spatio-temporal domain. The key component of the method is our new metric, CaMoJAB, which considers how the judder, aliasing, and blur artifacts introduced by the VRS at a given refresh rate are masked by hold-type blur, eye motion blur, and limited spatio-temporal sensitivity of the visual system. The metric is shown to explain multiple datasets with only a few fitted parameters.

We hope the proposed technique can improve the visual quality not only for regular adaptive-sync monitors, but also for VR headsets. Relatively high velocities of motion in those headsets have the potential to mask a portion of VRS artifacts. The adaptive choice of resolution and refresh rate could improve the quality of experience as the users would not be forced to chose one or another. Finally, the technique could reduce the rendering cost and extend battery life, especially when combined with foveated rendering.

ACKNOWLEDGMENTS

We would like to thank Joe March for his valuable comments. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement N° 765911 (RealVision) and under the European Research Council (ERC) Consolidator Grant agreement N° 725253 (EyeCode).

REFERENCES

Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. 2019. *Real-time rendering*. Crc Press.

AMD. 2021. AMD FidelityFX. https://www.amd.com/en/technologies/radeon-software-fidelityfx

Peter G. J. Barten. 2003. Formula for the contrast sensitivity of the human eye. In *Image Quality and System Performance*, Yoichi Miyake and D. Rene Rasmussen (Eds.), Vol. 5294. International Society for Optics and Photonics, SPIE, 231 – 238. https://doi.org/10.1117/12.537476

Mark R Bolin and Gary W Meyer. 1995. A frequency based ray tracer. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 409–418.

Kirsten Cater, Alan Chalmers, and Greg Ward. 2003. Detail to attention: exploiting visual tasks for selective rendering. In *ACM International Conference Proceeding Series*, Vol. 44. 270–280.

Alexandre Chapiro, Robin Atkins, and Scott Daly. 2019. A Luminance-aware Model of Judder Perception. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 1–10.

Roelof Roderick Colenbrander. 2021. Adaptive graphics for cloud gaming. US Patent App. 16/688,369.

Scott Daly, Ning Xu, James Crenshaw, and Vikrant J Zunjarrao. 2015. A psychophysical study exploring judder using fundamental signals and complex imagery. *SMPTE Motion Imaging Journal* 124, 7 (2015), 62–70.

James Davis, Yi-Hsuan Hsieh, and Hung-Chi Lee. 2015. Humans perceive flicker artifacts at 500 Hz. *Scientific reports* 5 (2015), 7861.

Kurt Debattista, Keith Bugeja, Sandro Spina, Thomas Bashford-Rogers, and Vedad Hulusic. 2018. Frame rate vs resolution: A subjective evaluation of spatiotemporal perceived quality under varying computational budgets. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 363–374.

Gyorgy Denes, Akshay Jindal, Aliaksei Mikhailiuk, and Rafał K. Mantiuk. 2020. A perceptual model of motion quality for rendering with adaptive refresh-rate and resolution. *ACM Transactions on Graphics* 39, 4 (jul 2020), 133. https://doi.org/10.1145/3386569.3392411

Michal Drobot. 2020. Software-based Variable Rate Shading in Call of Duty: Modern Warfare. In *ACM SIGGRAPH 2020 Courses*.

Xiao-Fan Feng. 2006. LCD motion blur analysis, perception, and reduction using synchronized backlight flashing. In *Human Vision and Electronic Imaging XI*. SPIE Vol. 6057, M1–14.

James A Ferwerda, Peter Shirley, Sumanta N Pattanaik, and Donald P Greenberg. 1997. A model of visual masking for computer graphics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 143–152.

Martin Fuchs, Tongbo Chen, Oliver Wang, Ramesh Raskar, Hans-Peter Seidel, and Hendrik P.A. Lensch. 2010. Real-time temporal shaping of high-speed video streams. *Computers & Graphics* 34 (2010), 575–584. Issue 5.

Alexander Goettker, Kevin J. MacKenzie, and T. Scott Murdison. 2020. Differences between oculomotor and perceptual artifacts for temporally limited head mounted displays. *Journal of the Society for Information Display* 28, 6 (jun 2020), 509–519. https://doi.org/10.1002/jsid.912

Gerben Johan Hekstra, Leo Jan Velthoven, and Michiel Adriaanszoon Klompenhouwer. 2008. Motion blur decrease in varying duty cycle. US Patent 7,317,445.

David M Hoffman, Vasiliy I Karasev, and Martin S Banks. 2011. Temporal presentation protocols in stereoscopic displays: Flicker visibility, perceived motion, and perceived depth. *Journal of the Society for Information Display* 19, 3 (2011), 271–297.

Intel. 2019. Use Variable Rate Shading (VRS) to Improve the User Experience in Real-Time Game Engines | SIGGRAPH 2019 Technical Sessions. https://www.slideshare.net/IntelSoftware/use-variable-rate-shading-vrs-to-improve-the-user-experience-in-real-time-game-engines

Adrian Jarabo, Tom Van Eyck, Veronica Sundstedt, Kavita Bala, Diego Gutierrez, and Carol O'Sullivan. 2012. Crowd light: Evaluating the perceived fidelity of illuminated dynamic scenes. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 565–574.

Paul V Johnson, Joohwan Kim, David M Hoffman, Andy D Vargas, and Martin S Banks. 2014. Motion artifacts on 240-Hz OLED stereoscopic 3D displays. *Journal of the Society for Information Display* 22, 8 (2014), 393–403.

Kimmo P Jokinen and Wen Nivala. 2017. 65-4: novel methods for measuring VR/AR performance factors from OLED/LCD. In *SID Symposium Digest of Technical Papers*, Vol. 48. Wiley Online Library, 961–964.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. 2004. *Multidimensional Knapsack Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 235–283. https://doi.org/10.1007/978-3-540-24777-7_9

Manuel Kraemer. 2018. Accelerating your vr games with vrworks. In *NVIDIAs GPU Technology Conference (GTC)*.

Yoshihiko Kuroki, Tomohiro Nishi, Seiji Kobayashi, Hideki Oyaizu, and Shinichi Yoshimura. 2007. A psychophysical study of improvements in motion-image quality by using high frame rates. *Journal of the Society for Information Display* 15, 1 (2007), 61–68.

James Larimer, Jennifer Gille, and James Wong. 2001. 41.2: Judder-Induced Edge Flicker in Moving Objects. In *SID Symposium Digest of Technical Papers*, Vol. 32. Wiley

Online Library, 1094–1097.
Chenglin Li, Laura Toni, Junni Zou, Hongkai Xiong, and Pascal Frossard. 2017. Delay-power-rate-distortion optimization of video representations for dynamic adaptive streaming. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 7 (2017), 1648–1664.
Peter Longhurst, Kurt Debattista, and Alan Chalmers. 2006. A GPU based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. 21–29.
Alex Mackin, Katy C Noland, and David R Bull. 2016. The visibility of motion artifacts and their effect on motion quality. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2435–2439.
Rafał K. Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. FovVideoVDP : A visible difference predictor for wide field-of-view video. *ACM Transaction on Graphics* 40, 4 (2021), 49. https://doi.org/10.1145/3450626.3459831
Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafal Mantiuk. 2020. Active Sampling for Pairwise Comparisons via Approximate Message Passing and Information Gain Maximization. In *2020 IEEE International Conference on Pattern Recognition (ICPR)*.
Karol Myszkowski. 1998. The visible differences predictor: Applications to global illumination problems. In *Eurographics Workshop on Rendering Techniques*. Springer, 223–236.
Karol Myszkowski, Przemyslaw Rokita, and Takehiro Tawara. 1999. Perceptually-Informed Accelerated Rendering of High Quality Walkthrough Sequences. In *Rendering Symposium*. 5–18.
Fernando Navarro, Susana Castillo, Francisco J Serón, and Diego Gutierrez. 2011. Perceptual considerations for motion blur rendering. *ACM Transactions on Applied Perception (TAP)* 8, 3 (2011), 1–15.
Nvidia. 2018. NVIDIA Turing GPU Architecture. https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf
A. Ortega and K. Ramchandran. 1998. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine* 15, 6 (1998), 23–50. https://doi.org/10.1109/79.733495
Maria Perez-Ortiz and Rafal K. Mantiuk. 2017. A practical guide and software for analysing pairwise comparison experiments. *arXiv preprint* (dec 2017). arXiv:1712.03686 http://arxiv.org/abs/1712.03686
Daniel Pohl, Timo Bolkart, Stefan Nickels, and Oliver Grau. 2015. Using astigmatism in wide angle HMDs to improve rendering. In *2015 IEEE Virtual Reality (VR)*. 263–264. https://doi.org/10.1109/VR.2015.7223396
Qualcomm. 2021. Qualcomm Snapdragon 888 levels up mobile gaming. https://www.qualcomm.com/news/onq/2020/12/02/qualcomm-snapdragon-888-levels-mobile-gaming
Mark Rejhon. 2017. Strobe Crosstalk: Blur Reduction Double-Images. https://blurbusters.com/faq/advanced-strobe-crosstalk-faq/
JE Roberts and AJ Wilkins. 2013. Flicker can be perceived during saccades at frequencies in excess of 1 kHz. *Lighting Research & Technology* 45, 1 (2013), 124–132.
Jan Schmid, Y ULUDAG, and J DELIGIANNIS. 2019. It just works: Raytraced reflections in" Battlefield V". In *GPU Technology Conference*.
T. Scott Murdison, Christopher McIntosh, James Hillis, and Kevin J. MacKenzie. 2019. 3-1: Psychophysical Evaluation of Persistence- and Frequency-Limited Displays for Virtual and Augmented Reality. *SID Symposium Digest of Technical Papers* 50, 1 (2019), 1–4. https://doi.org/10.1002/sdtp.12840 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sdtp.12840
BT Series. 2020. The present state of ultra-high definition television. https://www.itu.int/pub/R-REP-BT.2246-7-2020. (2020).
AAS Sluyterman. 2006. What is needed in LCD panels to achieve CRT-like motion portrayal? *Journal of the Society for Information Display* 14, 8 (2006), 681–686.
M. Stengel, P. Bauszat, M. Eisemann, E. Eisemann, and M. Magnor. 2015. Temporal Video Filtering and Exposure Control for Perceptual Motion Blur. *Visualization and Computer Graphics, IEEE Transactions on* 21, 2 (2015), 1–11. https://doi.org/10.1109/TVCG.2014.2377753
K. Sung, A. Pearce, and C. Wang. 2002. Spatial-Temporal Antialiasing. *IEEE Transactions on Visualization and Computer Graphics* 8, 2 (2002), 144–153.
Okan Tarhan Tursun, Elena Arabadzhiyska, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. 2019. Luminance-Contrast-Aware Foveated Rendering. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH)* 38, 4 (2019).
Unity. 2021. Unity - Manual: Dynamic resolution. https://docs.unity3d.com/Manual/DynamicResolution.html
Unreal. 2021. https://docs.unrealengine.com/en-US/RenderingAndGraphics/DynamicResolution/index.html
Karthik Vaidyanathan, Marco Salvi, Robert Toth, Tim Foley, Tomas Akenine-Möller, Jim Nilsson, Jacob Munkberg, Jon Hasselgren, Masamichi Sugihara, Petrik Clarberg, et al. 2014. Coarse pixel shading. In *Proceedings of High Performance Graphics*. 9–18.
Karthik Vaidyanathan, Robert Toth, Marco Salvi, Solomon Boulos, and Aaron E Lefohn. 2012. Adaptive Image Space Shading for Motion and Defocus Blur.. In *High Performance Graphics*. 13–21.
Tom Verbeure, Gerrit A Slavenburg, Thomas F Fox, Robert Jan Schutten, Luis Mariano Lucas, and Marcel Dominicus Janssens. 2017. System, method, and computer program product for combining low motion blur and variable refresh rate in a display. US Patent 9,773,460.
Andrew B Watson. 2013. High frame rates and human vision: A view through the window of visibility. *SMPTE Motion Imaging Journal* 122, 2 (2013), 18–32.
Andrew B Watson and Albert J Ahumada. 2011. Blur clarified: A review and synthesis of blur discrimination. *Journal of Vision* 11, 5 (2011), 10–10.
Martin Weier, Michael Stengel, Thorsten Roth, Piotr Didyk, Elmar Eisemann, Martin Eisemann, Steve Grogorick, André Hinkenjann, Ernst Kruijff, Marcus Magnor, et al. 2017. Perception-driven accelerated rendering. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 611–643.
Laurie M Wilcox, Robert S Allison, John Helliker, Bert Dunk, and Roy C Anthony. 2015. Evidence that viewers prefer higher frame-rate film. *ACM Transactions on Applied Perception (TAP)* 12, 4 (2015), 1–12.
Lance Williams. 1983. Pyramidal parametrics. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*. 1–11.
Himanshu Yadav and B Annappa. 2017. Adaptive GPU resource scheduling on virtualized servers in cloud gaming. In *2017 Conference on Information and Communication Technology (CICT)*. IEEE, 1–6.
Lei Yang, Dmitry Zhdan, Emmett Kilgariff, Eric B Lum, Yubo Zhang, Matthew Johnson, and Henrik Rydgård. 2019. Visually Lossless Content and Motion Adaptive Shading in Games. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 1 (2019), 1–19.
Hector Yee, Sumanta Pattanaik, and Donald P. Greenberg. 2001. Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments. *ACM Trans. Graph.* 20, 1 (2001), 39–65.

A EXPERIMENT 1: VRS MOTION QUALITY MEASUREMENT

In this appendix, we provide details of the setup and the procedure used in our motion quality experiment described in Section 4.3.

A.1 Setup

To study the effect of shading rate reduction on motion quality, observers were shown side-by-side two animations at different shading rates and textures on two separate displays (Figure 14). The experiment was divided into three blocks, each with a different configuration, simulating different display devices (Table 3). For the first two blocks, we used a pair of ASUS ROG Swift PG279Q 27" WQHD displays as they were capable of showing a wide range of refresh rates and luminance. For the last block, which studied low persistence, we simulated two virtual displays in a VR headset (Valve Index). We were unable to use the low persistence mode (ULMB) provided by our ASUS display as it led to excessive ghosting caused by strobe cross-talk [Rejhon 2017]. The ASUS displays were calibrated with JETI Specbos 1211-2 spectroradiometer to achieve an accurate luminance reproduction and placed side-by-side to facilitate the comparison of horizontal motion. The experiment was designed in Unity3D and we used VRS offered by Nvidia RTX 2080Ti for shading rate reduction. To prevent observer fatigue, we ensured that the experiment time did not exceed 30 minutes.

A.2 Stimuli

Based on the previous work detailed in Section 2, we identified the following factors to have the highest effect on the quality of motion: refresh rate, persistence, resolution, luminance, contrast, object velocity, and shading rate. Due to the high dimensionality of this problem, we restricted our experiment to three blocks (regions

Fig. 14. Experiment Setup. We used a chinrest to keep viewer's head at a fixed location.

of the space) according to what we believe to be the typical rendering configurations of our target application, namely, PC, mobile, and VR real-time rendering. The exact details of the range of each tested dimension can be found in Table 3. The resolution was varied by changing the viewing distance or scaling the viewport. For content, we used textured 3D models of simple shapes (cube, cone, sphere, cylinder, and capsule) and two popular more complex models (Stanford bunny and Utah teapot). We used 4 textures: checkerboard, linear-gradient, salt-and-pepper noise, and grass, shown in Figure 7a of the main paper. The textures and contrast were chosen to bring out the most common spatial and temporal artifacts associated with shading rate reduction and to include a wide range of variations in spatial frequency content. The textures were pre-processed to produce the selected luminance and contrast levels for each block. 2× supersampling anti-aliasing (SSAA) was used only in VR (Config3) to compensate for low PPD and to reduce the visibility of the artifacts. All the models moved horizontally at the same velocity. The velocities in each block were selected to be representative of common velocities in real-time graphics on respective displays. A small rotational velocity (<2 deg/s) was added to each model randomly so that all parts of the model were visible over the experiment. Since there was no vertical motion in our stimulus, we used the only possible combination of available shading rates that would keep the vertical rate constant while varying the horizontal shading rate between $\{1 \times 2, 2 \times 2, 4 \times 2\}$[7]. The experiment was divided into 3 separate sessions, one for each block.

A.3 Procedure

13 participants aged 21–40, 4 females and 9 males with normal or corrected-to-normal vision participated in the experiment. The experiment used a pairwise comparison protocol for quality assessment. In each trial, the participants were presented with two moving stimuli at different shading rates and potentially different textures. They had an option to see the reference stimuli, rendered at the original resolution (VRS 1×1) by pressing and holding the space bar. A short blank of 0.5 s was shown before switching between test and reference animations so that the participants could not use temporal flicker to detect the presence of artifacts. Participants were then asked to pick the stimulus closer to its respective reference (stimulus with fewer artifacts). Before the experiment, all participants were briefed in a short training session where we highlighted the motion artifacts caused by VRS. Since we were limited to a relatively small number of participants due to COVID-19 restrictions, we collected more data from each participant to reduce confidence intervals. Each participant performed 120 comparisons for each of the three blocks (4680 comparisons across all participants). The order of comparisons was determined using an active sampling method [Mikhailiuk et al. 2020] to maximize the information gained through each trial.

[7] A shading rate of ×N means 1/N shading resolution