

Enhancing Foveated Rendering with Weighted Reservoir Sampling

Ville Cantory
University of Minnesota
Minneapolis, MN, USA
canto063@umn.edu

Darya Biparva
University of Minnesota
Minneapolis, MN, USA
bipar001@umn.edu

Haoyu Tan
University of Minnesota
Minneapolis, USA
tan0021@umn.edu

Tongyu Nie
University of Minnesota
Minneapolis, USA
nie00035@umn.edu

John Schroeder
University of Minnesota
Minneapolis, USA
schr1294@umn.edu

Ruofei Du
Google
San Francisco, CA, USA
ruofei@google.com

Victoria Interrante
University of Minnesota
Minneapolis, MN, USA
interran@umn.edu

Piotr Didyk
Università della Svizzera italiana
Lugano, Switzerland
piotr.didyk@usi.ch

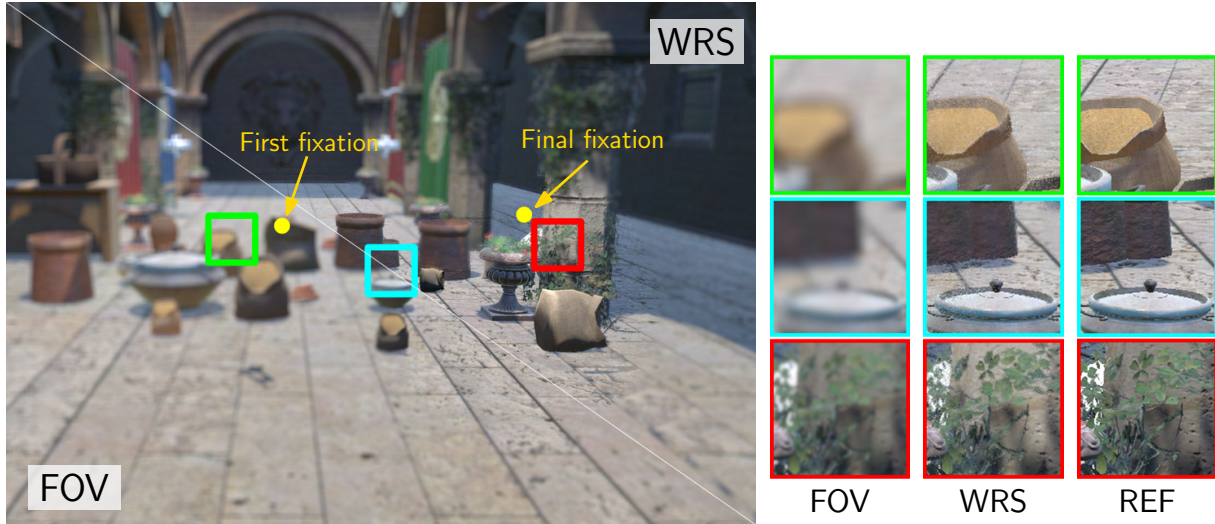


Figure 1: Our method temporally enhances the output of foveated rendering. Given an eye-tracked scanpath, traditional rasterization-based foveated rendering (FOV) methods (left) do not preserve previously rendered foveal samples. Our enhancement method (right) uses Weighted Reservoir Sampling (WRS) to stochastically preserve high-quality rasterized samples across time, leading to higher overall image quality. The images above feature a 5° fovea and were designed to be viewed at 70° field-of-view. Sponza courtesy of Casual Effects [McGuire 2017].

Abstract

Spatiotemporal sensitivity to high frequency information declines with increased peripheral eccentricity. Foveated rendering exploits

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MIG '25, Zurich, Switzerland

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2236-3/25/12
<https://doi.org/10.1145/3769047.3769058>

this by decreasing the spatial resolution of rendered images in peripheral vision, reducing the rendering cost by omitting high frequency details. As foveation levels increase, the rendering quality is reduced, and traditional foveated rendering systems tend not to preserve samples that were previously rendered at high spatial resolution in previous frames. Additionally, prior research has shown that saccade landing positions are distributed around a target location rather than landing at a single point, and that even during fixations, eyes perform small microsaccades around a fixation point. This creates an opportunity for sampling from temporally neighbouring frames with differing foveal locations to reduce the required rendered size of the foveal region while achieving a higher perceived

image quality. We further observe that the temporal presentation of pixels frame-to-frame can be viewed as a data stream, presenting a random sampling problem. Following this intuition, we propose a Weighted Reservoir Sampling technique to efficiently maintain a reservoir of the perceptually relevant high quality pixel samples from previous frames and incorporate them into the computation of the current frame. This allows the renderer to render a smaller region of foveal pixels per frame by temporally reusing pixel samples that are still relevant to reconstruct a higher perceived image quality, while allowing for higher levels of foveation. Our method operates on the output of foveated rendering, and runs in under 1 ms at 4K resolution, making it highly efficient and integrable with real-time VR and AR foveated rendering systems.

CCS Concepts

• **Computing methodologies** → **Rasterization; Virtual reality; Computer graphics.**

Keywords

foveated rendering, weighted reservoir sampling

ACM Reference Format:

Ville Cantory, Darya Biparva, Haoyu Tan, Tongyu Nie, John Schroeder, Ruofei Du, Victoria Interrante, and Piotr Didyk. 2025. Enhancing Foveated Rendering with Weighted Reservoir Sampling. In *The 18th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '25), December 03–05, 2025, Zurich, Switzerland*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3769047.3769058>

1 Introduction

Real-time rendering requirements continue to grow as shading model accuracy and spatiotemporal display standards increase. The top of the line consumer VR headset, the Varjo XR-4¹ boasts a resolution approximately 3.5× what a traditional 4K (3840 × 2160) resolution displays. Efficient rendering for these displays remains challenging, necessitating the development of ways to lower computational costs, ideally without an impact on perceptual quality.

Foveal vision is an approximately 2° region of vision with high visual acuity, with the rest of the visual field suffering from lower visual acuity [Bergmann 1854; Thibos et al. 1996]. To save computational resources, foveated rendering takes advantage of the reduction in visual acuity by rendering lower quality samples in regions that diverge from the fovea. This is commonly done by reducing the sampling rate in peripheral vision, eliminating high-spatial frequency details in peripheral portions of the scene; yet it has been consistently observed that the amount of foveation tenable depends on the scene viewed, with some modulators for this including the frequency content of the image viewed [Patney et al. 2016], luminance [Tursun et al. 2019], attention subtended to different visual fields [Krajancich et al. 2023], and more. It has also been shown that the presence of high frequency details, even in peripheral vision, can be detected and are important for visual quality, with some works opting to reconstruct image metamers – high frequency details that match the statistics of the scene content, but do not need to contain the same content as the original ground

truth in order to have a high perceptual quality [Tariq et al. 2022; Walton et al. 2021].

Most current approaches fail to persist previously rendered high-resolution samples, replacing them with lower resolution samples as gaze location shifts. If a fragment were previously rendered at a high spatial resolution, and visibility is maintained but the fragment is not otherwise changed due to occlusions or shading changes, replacing it with a lower quality version is unnecessary. Discarding recently rendered high spatial resolution samples also overlooks common oculomotor behaviour; Saccades rarely land precisely on their intended target [Van der Stigchel et al. 2020], and are corrected with microsaccades and ocular drift. Further, fixations themselves are interspersed with microsaccades rather than truly fixating at a single location [Van Beers 2007; Zuber et al. 1965]. The Human Visual System (HVS) integrates information temporally, meaning transient high-resolution details can contribute to the final image perception. Thus, discarding high-quality samples immediately upon gaze shifts may discard perceptually relevant details.

Naturally, this creates an analogue to a data streaming problem, where perceptually important pixel samples arrive over many frames and must be selectively retained. Reservoir sampling is a family of sampling algorithms designed to maintain a representative subset of elements from a stream without preprocessing the stream [Fan et al. 1962], and was recently introduced to computer graphics for sampling light paths for real-time-raytracing (ReSTIR) [Bitterli et al. 2020].

Similar to ReSTIR, we employ *Weighted Reservoir Sampling* (WRS), a variant of reservoir sampling in which each stream element is assigned a weight w drawn from a distribution p representing its importance. However, we focus on foveated rasterization and temporally accumulating higher quality samples from a foveated data stream, prioritizing samples by perceptual relevance.

We introduce a novel temporal bias function to bias the reservoir based on our interpretation of the reservoir sampling process as a Bernoulli Trial to prevent stale samples from inappropriately persisting in the reservoir. Our method significantly reduces the number of foveal pixels that have to be rendered per-frame while maintaining high perceptual quality.

Our primary contributions are:

- Introducing Weighted Reservoir Sampling (WRS) to foveated rasterization to temporally accumulate high-quality pixel samples;
- Proposing a new temporal bias function based on Bernoulli Trials to reweight reservoir samples without dependence on absolute or relative time that samples were selected for inclusion in the reservoir;
- Validation of our approach via user studies to improve foveated text reading speed, and improved image quality over foveated rendering and foveated rendering with Temporal Antialiasing (TAA).

2 Related Work

Our work is inspired by a wide body of prior research in foveal and peripheral vision and foveated rendering. In this section, we introduce prior perceptual work about foveal and peripheral vision in § 2.1. We proceed to discuss a body of previous work in foveated

¹Varjo XR-4: <https://varjo.com/products/xr-4/>

rendering in § 2.2, and prior temporally adaptive rendering work in § 2.3

2.1 Perceptual Background

2.1.1 Foveal and Peripheral Vision. The Human Visual System (HVS) features a 2° central region of vision capable of high visual acuity, with the rest of the visual field perceiving progressively lower visual acuity [Aubert and Foerster 1857; Lettvin 1976; Strasburger et al. 2018]. The physiological reason for this is there exists a region of the eye with a dense field of photoreceptors (cones), deemed the *fovea centralis* [Bergmann 1854, 1857]; the rest of the eye is less densely populated with photoreceptors (mostly rods) resembling a blue noise distribution [Curcio and Allen 1990; Curcio et al. 1990; Lanaro et al. 2020]. While there is a rapid optical decrease in visual acuity and retinal ganglion-cell density as eccentricity from the fovea increases [Watson 2014], neural factors degrade visual acuity more rapidly [Navarro et al. 1993; Williams et al. 1996]. Moreover, this can lead to a range of spatial frequencies in peripheral vision for which certain frequencies are detectable, but low-level details are not resolvable [Sere et al. 2000; Thibos et al. 1987].

2.1.2 Saccades. Saccades are large eye movements [Javal 1878] under which visual sensitivity is reduced [Erdmann and Dodge 1898; Richards 1969], producing retinal blur during eye movements [Dodge 1900; Volkmann 1962]. Saccades typically occur 2-4 times per second [Otero-Millan et al. 2013] and last under 50ms [Robinson 1964]. While a saccade can subtend up to 200° [Robinson 1964], a *microsaccade* often moves around 1° of visual angle during fixation [Zuber et al. 1965]. Microsaccades can enhance the perception of fine details during fixation [Rucci et al. 2007]; after a saccade from one fixation point to another, many small ocular drifts and microsaccades can involuntarily follow [Rucci and Poletti 2015]. Visual acuity is not just reduced during the saccade, but for 50-500ms after saccadic landing (completion of the saccade) depending on the spatial frequency of the content [Kwak et al. 2024].

In our work, we seek to exploit microsaccades by utilizing them to expand the effective fovea size displayed to the user.

2.2 Foveated Rendering

Foveated rendering achieves computational speedups by exploiting the reduction in peripheral visual acuity. Early works on foveated rendering focused on using image and video compression to improve bandwidth [Kortum and Geisler 1996; Tsumura et al. 1996]. Gaze-contingent mesh simplification methods have also been thoroughly explored [Ohshima et al. 1996; Surace et al. 2023]; however, as fragment shading tends to be much more computationally expensive than geometry processing [He et al. 2014], most efforts seeking performance benefits focus on reducing shading costs.

Eccentricity-dependent shading rate reductions, driven by the sharp drop in visual acuity as retinal eccentricity increases, has been the focus of much modern foveated rendering research. Commonly used thresholds for the detection of spatial frequency reductions (accomplished with reductions in fragment shading rates) were first formalized in Guenter et al. [2012], and further improved in subsequent works [Patney et al. 2016; Vaidyanathan et al. 2014]. Similarly, Meng et al. [2018] achieved higher performance benefits

by utilizing a 2-pass foveated rendering pipeline specifically built for deferred shading. As the amount of foveation tolerable depends on many factors such as eye motion, contrast, luminance, image content, attentional load, and more [Albert et al. 2017], there have been some explorations into finding better adaptive sampling patterns to drive a reduction in peripheral shading rates based on these factors [Stengel et al. 2016; Tursun et al. 2019], resulting in sample count reductions in raytracing and path tracing [Polychronakis et al. 2021; Swafford et al. 2016; Weier et al. 2016] and optimizations in split-rendering systems [Cantory and Ringo 2023].

Recent works have added high frequency details to peripheral regions to better preserve the perceptual experience of full-resolution rendering under foveation. Walton et al. [2021] constructed foveated image metamers - images that are different from the ground truth, but will be perceived as the same - and Tariq and Didyk [2024] explored motion metamerism. Tariq et al. [2022] incorporated adding back in Gabor noise to achieve higher levels of foveation, offset by the noise metamer. DeepFovea [Kaplanian et al. 2019] utilized Generative Adversarial Networks [Goodfellow et al. 2014] to re-construct foveated images with extremely sparse sampling. For a comprehensive review of foveated rendering, we refer readers to [Wang et al. 2023].

Outside of raytracing and pathtracing, we are unaware of any prior gaze-contingent rendering works stochastically accumulating well-sampled pixels to temporally increase image quality. Within a rasterization context, it stands to reason that if only the fixation point and not the viewport have drastically changed frame to frame, pixels that were previously rendered at a high shading rate can be reused to expand the number of well-sampled pixels without unduly increasing the cost to render them. In our work, we utilize a Reservoir Sampling algorithm to achieve higher quality temporal image reconstruction from a stream of foveated images.

2.3 Temporally Adaptive Rendering

Temporal reuse to achieve higher image quality has been widely used in computer graphics applications. Intuitively, most applications change very little in the 8-16 ms that occurs between frames, meaning that much of the information computed from previous frames can still be relevant to the current one, achieving possibly an increased framerate by reducing the rendering cost, or improving the image quality of each frame.

The Reverse Reprojection Cache (RRC) proposed by Nehab et al. [2007] and Scherzer et al. [2007] for reprojection with shadow maps, describes a framebuffer that contains pixel data visible at a given frame, and the scene depth in screen space rendered from the camera's point of view. RRC's can be used to determine the location where a pixel was in the previous frame, as well as the visibility. However, non-linear warping often incurs cache misses due to visibility changes or view-dependent shading changes. Re-rendering or inpainting can occur [Marroquin et al. 2007], but cheaper methods of shading reuse have been explored using a *compact geometry buffer* that stores and reuses shading samples separate from visibility, [Liktov and Dachsbacher 2012], or through methods that adaptively change the shading rate using error estimation based on spatial frequency analysis [Yang et al. 2019].

A common usage of temporal reuse in consumer applications has been Temporal Antialiasing (TAA). Originally introduced by Korein and Badler [1983] to reduce temporal aliasing, modern TAA methods often seek to reduce spatial aliasing by using temporal samples to achieve supersampling. TAA reuses subpixel samples accumulated from previous frames to achieve supersampling, generally relying on jittered subsamples to be generated each frame. This leads to much better integration over a pixel's domain, and the samples are summed using a weighted average and stored in a history buffer representing the samples that have been rendered. Reverse reprojection is used to map samples to their previous location; however, when the history buffer is invalidated by something like large shading changes or occlusions / disocclusions, the history buffer may be invalidated at an affected pixel, resetting the effective sample count.

Temporal Upscaling (TAAU) seeks to reduce the sample rate per pixel to under one per pixel for the target resolution. It accumulates lower resolution shading results to produce a higher resolution over many frames. However, as noted by Yang et al. [2020], ghosting and artifacts can occur in TAA solutions if the history buffer is not carefully rectified when it may become invalidated. More recently, DLSS has used deep neural networks to achieve fast supersampling, relying on reverse reprojection techniques and reusing the previous frame to supersample the next frame. For a comprehensive overview of temporal reuse, we refer readers to Scherzer et al. [2011]; Yang et al. [2020].

3 Mathematical Preliminaries

This section outlines the mathematical basis of our reservoir sampling method and highlights prior applications of weighted reservoir sampling in computer graphics.

3.1 Weighted Reservoir Sampling

Reservoir sampling is a technique to efficiently sample n items from a population of potentially unknown size N . Originally introduced by Fan et al. [1962] for sequential sampling from tape drives in a single pass, reservoir sampling was later formalized for uniform sampling from data streams [Vitter 1984, 1985]. It has also been generalized to allow non-uniform sampling probabilities, leading to Weighted Reservoir Sampling (WRS) [Chao 1982]. This variation is particularly well-suited to cases where evaluating large sets of candidate samples is expensive and real-time performance is required. It can also be applied to sampling problems where the input data is evolving [Aggarwal 2006; Efraimidis 2015]. These characteristics make WRS an attractive choice for computer graphics applications, where large and expensive sets of candidate samples, such as light paths or pixel contributions, must be considered under tight computational budgets. In such cases, WRS enables efficient sampling from a dynamically evolving stream in a single pass, avoiding repeated evaluations of the full set. In particular, WRS enables sampling from a probability density function, p , where each candidate's weight reflects its relative importance. Because p is often expensive to exactly evaluate in most rendering contexts, samples are typically drawn from an approximation, $\hat{p}$ [Bitterli et al. 2020; Ouyang et al. 2021; Wyman and Panteleev 2021].

Formally, given a stream of candidate samples $\mathbf{x}_1, \dots, \mathbf{x}_N$, each associated with a positive weight $w_i \propto \hat{p}(\mathbf{x}_i)$, the goal of WRS is to maintain a representative sample of size n from this stream. For the remainder of this paper, we focus on cases where the reservoir operates under tight memory constraints, and thus we maintain a reservoir containing only a single sample ($n = 1$) and that the total length of the data sequence, N , is unknown. The items of the population are not assumed to be unique, but they are distinguishable.

The core algorithms for reservoir sampling are A-Chao [Chao 1982] and A-ES [Efraimidis 2015; Efraimidis and Spirakis 2006]. In A-Chao, when a new sample $\mathbf{x}_i$ is encountered, its selection probability is computed and used to decide whether it replaces the current reservoir entry, as:

$$p(\text{select } \mathbf{x}_i) = \frac{w_i}{\sum_{j=1}^i w_j}. \quad (1)$$

The key insight is that if reservoir items are sampled proportionally to their weights, then uniformly replacing one preserves this distribution. We present a modified version of A-Chao for a reservoir of size $n = 1$ in Algorithm 1.

Algorithm 1 A-Chao Update Reservoir

```

1: function UPDATE( $\mathbf{x}_i, w_i$ )
2:    $w_{\text{sum}} \leftarrow w_{\text{sum}} + w_i$ 
3:   if  $\text{rand} \cdot w_{\text{sum}} < w_i$  then
4:      $\mathbf{x} \leftarrow \mathbf{x}_i$ 
5:      $w \leftarrow w_i$ 
6:   end if
7: end function

```

In this formulation, it is assumed that a reservoir structure maintains three quantities: the current sample in the reservoir $\mathbf{x}$, its weight w , and the cumulative weight sum w_{sum} of all observed candidates. $\mathbf{x}_i$ is the sample that is being evaluated and $\text{rand} \sim \mathcal{U}[0, 1]$. This allows the sampling decision to be made with minimal memory requirements.

Algorithm A-ES introduced by Efraimidis and Spirakis [2006] takes a different approach. They proposed a sampling schema dependent on a random key, $k_i = \text{rand}_i^{1/w_i}$ for stochastically updating a reservoir item. The reservoir, at each step, will contain n largest keys from a weighted random stream. Larger weights produce larger denominators in the exponent. They argue that a uniform random variable can be “amplified” by raising it to an appropriate power. In this formulation, it is assumed that reservoir structure maintains the sample $\mathbf{x}$ and the key for this sample k , but the weighted sum of all previously evaluated candidates is not stored, making it more difficult to temporally evaluate the sample quality.

The aforementioned sampling algorithms focus on unbiased sampling; yet to support cases of reservoir sampling from an evolving data stream, a temporal bias can be applied either into the weight of items currently in the reservoir, or into the weight of newly evaluated items [Efraimidis 2015]. A commonly used bias function is the *memory-less* temporal bias introduced by Aggarwal [2006]:

$$f(j, i) = e^{-\lambda(i-j)}. \quad (2)$$

It assumes uniform sampling with equal weights w_i and represents an exponential *damping function* [Diop et al. 2024], where λ is

a predefined bias rate, j is the index of the sample currently in the reservoir (x), and i is the index of the sample currently being evaluated against x . Given a bias function $f(j, i)$, samples in the reservoir can be reweighted such that more recently evaluated samples are more likely to be selected for inclusion: $w = w \cdot f(j, i)$. Memory-less bias functions reduce the probability of each reservoir sample to persist in the reservoir as relative time passes, but the absolute time of insertion / comparison does not matter.

3.2 WRS in Computer Graphics

WRS combined with resampled importance sampling (ReSTIR) has been extensively used to evaluate real-time raytracing ray queries [Bitterli et al. 2020; Wyman et al. 2023]. ReSTIR spatiotemporally resamples candidate light samples to progressively increase the sample quality that leads to faster and higher quality real-time raytracing. Their version of WRS uses replacement, meaning that the population of candidates to be evaluated is unaltered after an item is selected. This version of reservoir sampling is usually done with Algorithm A-Chao [Efrimidis 2015; Meligrana 2024].

They contend that WRS enables efficient spatiotemporal reuse by allowing for spatiotemporally neighbouring reservoirs to be combined without needing to reevaluate each sample distribution. Since a reservoir encodes its selected sample, sample weight, and the cumulative sum of all evaluated candidates, combining reservoirs reduces to treating each reservoir as a single weighted sample. For instance, combining reservoir R_1 that has seen m_1 candidates with neighbour R_2 that has seen m_2 candidates is equivalent to performing WRS on a stream of two samples. This preserves the invariant of WRS and increases the effective sample count to $m_1 + m_2$. When combining h reservoirs, the sample count increases to $m_1 + \dots + m_h$.

We show the Combine Reservoirs algorithm introduced by ReSTIR [Bitterli et al. 2020] in Algorithm 2 without resampling against differing target distributions. As it is most common to compare only two reservoirs at a time (i.e., compare a current pixel's reservoir to its temporal neighbour or to a spatial neighbour), the presented algorithm is shown in simplified form.

Algorithm 2 Combine Reservoirs

```

1: function COMBINE( $R_1, R_2$ )
2:    $R_{\text{out}} \leftarrow \begin{cases} R_1 & \text{if } (\text{rand} \cdot (R_1.w_{\text{sum}} + R_2.w_{\text{sum}}) \leq R_1.w_{\text{sum}}) \\ R_2 & \text{otherwise} \end{cases}$ 
3:    $R_{\text{out}.w_{\text{sum}}} \leftarrow R_1.w_{\text{sum}} + R_2.w_{\text{sum}}$ 
4:    $R_{\text{out}.m} \leftarrow R_1.m + R_2.m$ 
5:   return  $R_{\text{out}}$ 
6: end function

```

4 Method Overview

Our main contribution is to introduce foveated rasterization as a reservoir sampling problem with temporal sample accumulation, enabling us to reduce the rendered size of the fovea per generated frame while temporally increasing the number of high-quality pixels displayed to the user. While there exist reservoir sampling approaches for real-time raytracing [Bitterli et al. 2020], rasterization poses different problems: We are not sampling potential light

paths and trying to select the best one to trace. Instead, each frame, we are given a rasterized pixel that has a given level of foveation applied to it, and the problem becomes whether to include that pixel in the final output displayed to a user. With an input of a sequence of foveated images, we demonstrate how the reservoir allows us to adaptively accumulate high-quality temporal samples.

4.1 WRS for Foveated Rendering

For each pixel location, the sequence of incoming values across frames forms a data stream, where each frame contributes a new candidate sample. The weight w_i associated with each candidate reflects its spatial proximity to the current gaze location, prioritizing samples that were rendered with greater visual fidelity this frame. Our method applies WRS independently at each pixel location, treating each as a separate stream. The reservoir retains an estimate of the most representative rendered sample observed so far for that pixel location, probabilistically determined by the weight of each candidate compared to the weighted sum of all previously evaluated candidates. Consequently, each frame consists of temporally accumulated samples that preserve perceptual relevance.

A proposed distribution $\hat{p}(x)$ from which weights are drawn should be efficient to evaluate and sample from while providing an approximation of the target distribution $p(x)$. Visual acuity thresholds for a given stimulus depend on a variety of factors such as luminance, spatial frequency of the stimuli, contrast, and more. Several thorough contrast sensitivity functions (CSFs) provide models to approximate this [Ashraf et al. 2024; Cai et al. 2024; Krajancich et al. 2021; Mantiuk et al. 2022], but computing the parameters to evaluate them becomes more expensive. Instead, we derive our $\hat{p}$ by sampling from the distribution of retinal cone density modeled by Watson [2014] on data from [Curcio and Allen 1990; Curcio et al. 1990] given in Equation (3).

$$d(e) = 2d_c(0) \left(1 + \frac{e}{r_m}\right)^{-1} \cdot \left[a_k \left(1 + \frac{e}{r_{2,k}}\right)^{-2} + (1 - a_k) \exp\left(-\frac{e}{r_{e,k}}\right) \right] \quad (3)$$

We use the parameters reported by Watson [2014]: $a_k = 0.9851$, $r_{2,k} = 1.058$, $r_{e,k} = 22.14$, $d_c(0) = 14804.6$, and e denotes the eccentricity. We then normalize Equation (3) by the peak retinal cone density at the edge of our fovea bounds if outside the fovea, or assign a weight of 1 if inside, producing Equation (4), and visualized in Figure 2.

$$\hat{p}(x) = w(e) = \begin{cases} 1, & \text{if } e \leq r_f \\ \frac{d(e)}{d(r_f)}, & \text{if } e > r_f \end{cases} \quad (4)$$

The two core algorithms introduced in § 3 differ subtly in how they apply the given weights, which makes them suitable for different types of applications. Algorithm A-Chao (1), is better suited for sampling from foveated streams because it maintains a weighted accumulation of all previously seen samples, allowing for more temporal continuity. When the total weight is high, it indicates that multiple high-quality samples have been likely observed, making it more likely that the current reservoir content reflects a sample with high spatiotemporal fidelity. In contrast, Algorithm A-ES considers only the weight of the current candidate relative to the sample

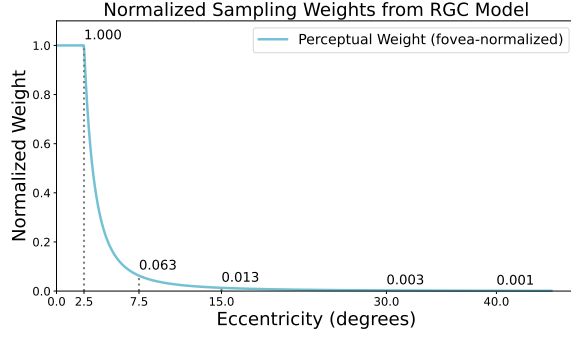


Figure 2: Visualization of Equation (4) weights. The distribution is drawn from an approximation of visual acuity falloff given by Watson [2014], normalized by the peak retinal ganglion cell density at a 2.5° eccentricity from the fovea.

already in the reservoir, without retaining any information about prior observations. Consequently, it lacks a mechanism for historical integration (for $n < N$) and is prone to discarding valuable high-quality samples too quickly. We validated this intuition empirically and observed that A-ES indeed tends to lose such samples more frequently than Algorithm 1. We note the observation from Bitterli et al. [2020], that since our sampling involves temporal accumulation, we can interpret the process as repeatedly combining two reservoirs: one reservoir R_1 , representing the accumulation of all previously seen samples up to the current frame i , and another reservoir R_2 containing just the newly rendered candidate and its associated weight. As detailed in § 3.2, combining two reservoirs reduces to treating each as a single weighted sample, while reaping the sampling benefits from the history the reservoir has evaluated. This observation allows us to use Algorithm 2 as proposed in Bitterli et al. [2020].

4.2 Temporal Biasing with Bernoulli Trials

As previously discussed, Algorithm A-Chao is more appropriate for our application because it accumulates sample weights over time. However, this same characteristic can become a liability in the presence of a rapidly evolving data stream. When the scene changes significantly, the algorithm may continue to give undue importance to previously observed samples that are no longer representative, leading to the rejection of new samples that better reflect the current state. This is a well-studied problem in reservoir sampling.

Temporal bias functions for reservoir sampling were introduced by Aggarwal [2006], which proposes a memory-less bias function to weight reservoir samples lower, such that newer samples have a higher probability to be selected for the reservoir at the time of their evaluation. This temporally biases reservoir samples based on the relative *time of sampling* compared to when new samples are evaluated. However, this requires solving for a λ that fits the general-case for which the reservoir sampling algorithm is designed, where λ is generally very small. In complex scenes, there is scarcely a constant λ that would fit all possible situations.

Rather than relying on fixed temporal decay, our goal is to ensure that significant scene changes are not ignored due to reservoir

inertia, while still preventing low-quality samples from prematurely overwriting important ones that are still perceptually valid. An alternative solution would be to deterministically preserve the highest weighted sample rendered across frames; however, this strategy assumes that high-fidelity samples will always remain valid or perceptually accurate. Occlusions / disocclusions, lighting changes, shading rate changes, and filtering effects can all alter the colour of any given pixel; in the case of an occlusion, the scene has fundamentally changed, while for a shading rate reduction, the scene has not changed. Moreover, diagnosing the specific cause of the reduction can be difficult and expensive, potentially imposing other uncertainty if we were to rely exclusively on the presence of a colour change [Mueller et al. 2021]. A deterministic reuse policy would therefore require a robust mechanism to discard samples that have become outdated due to scene or view-dependent changes. Tuning this mechanism to perform well across the wide range of conditions encountered in immersive environments may incur high complexity and cost; thus, a probabilistic formulation that naturally accounts for this variability and uncertainty could prove more appropriate for high performance situations.

We propose biasing them proportionally to the chance that they survived or not. As discussed previously, if $\frac{w_i}{R_1 \cdot w_{sum} + w_i} > rand$, then the current sample held in the reservoir will be replaced; otherwise, it will be maintained in the reservoir. Therefore, the chance that the sample currently in the reservoir survives is approximately:

$$\hat{p}(\text{survive}) = 1 - \frac{w_i}{R_1 \cdot w_{sum} + w_i} \quad (5)$$

This model of whether a sample survives the update comparison is a Bernoulli trial, and reservoir samples are repeatedly biased over time using this, with w_i and w_{sum} changing frame-to-frame. In the case where R_1 is the previous frame’s reservoir (the *temporal* reservoir) and R_2 is the reservoir for the newly rendered sample (and thus $w_i = R_2 \cdot w_{sum}$), we can bias $R_1 \cdot w_{sum}$ with the survival rate when it will be compared to R_2 ’s sample, giving:

$$R_1 \cdot w_{sum} = R_1 \cdot w_{sum} \cdot \left(1 - \frac{R_2 \cdot w_{sum}}{R_1 \cdot w_{sum} + R_2 \cdot w_{sum}} \right) \quad (6)$$

Over k frames, this becomes a Binomial distribution with varying probabilities of survival for every $i < k$ iteration. After R_1 is biased according to the incoming probability of the new sample in R_2 , we call the combine function in Algorithm 2 to sample the two reservoirs and output a final selected reservoir with an estimate for the best quality pixel sample.

4.3 Implementation

We implement our WRS method with temporal biasing using the C# and HLSL in the Built-In Rendering Pipeline in Unity3D 6.0. Foveated rendering is simulated as a postprocess with two methods: The first uses simple mipmapping with bilinear filtering of the output framebuffer, similar to the shading rate falloffs proposed in Patney et al. [2016], and the second uses a progressively increasing Gaussian blur, with blur parameters derived from Tursun et al. [2019]. In raytracing or pathtracing, a lower proportion of peripheral pixels would be accurately estimated, running at a low sampling accuracy per pixel area. In our rasterizer, shading is performed at a reduced perceptual quality to simulate foveated rendering by

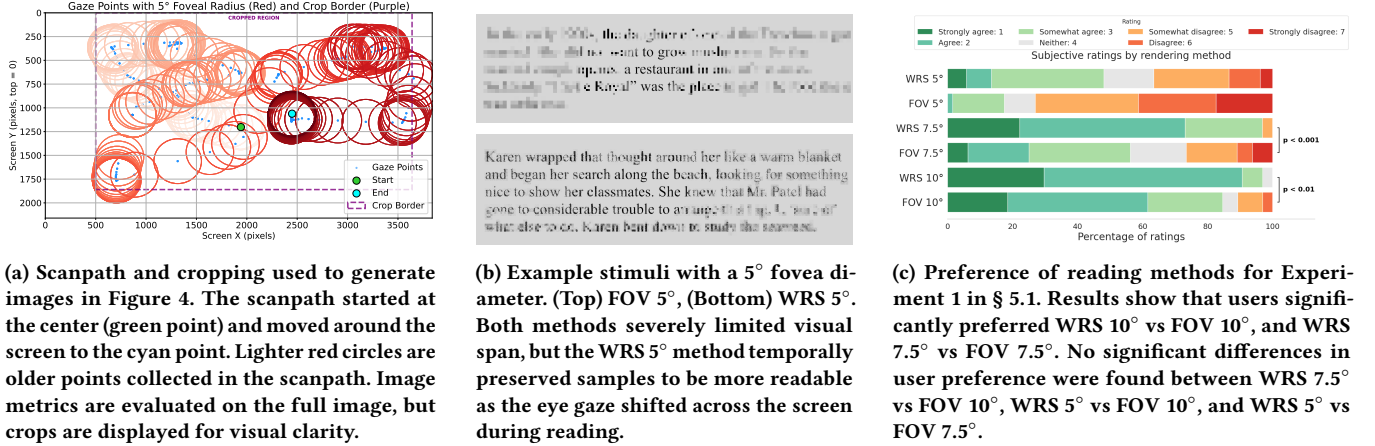


Figure 3

computing mipmaps and performing filtering operations. This is equivalent to rendering foveated pixels into a reduced resolution buffer and then interpolating pixel values.

Temporal reprojection using motion vectors is used to compare temporally neighbouring reservoirs after scene or camera movement has occurred. Depth buffer comparisons between frames are used to flush a reservoir’s temporal sample if an occlusion or disocclusion takes place. Finally, perceptual colour changes $\Delta L = \text{luminance}_x - \text{luminance}_{xi}$ are used in our temporal bias function by converting the pixel for the current frame and the temporal neighbour’s reservoir sample to LAB colour space (Delta E 94), and then applying the *full* temporal bias function:

$$R_{1wsum} = R_{1wsum} \cdot (1 - |\Delta L|) \cdot \left(1 - \frac{R_{2wsum}}{R_{1wsum} + R_{2wsum}}\right) \quad (7)$$

5 Evaluation

To evaluate our method, we conducted a series of human subject experiments to understand what impact temporally accumulating foveated samples may have on common user experiences. The experiments were conducted following a study protocol that was reviewed and approved by our Institutional Review Board (IRB). We also compare WRS runtime performance and perceptual image metrics (PSNR, SSIM, LPIPS, FovVideoVDP [Mantiuk et al. 2021]) against standard foveated rendering, and standard foveated rendering with TAA applied.

5.1 Experiment 1: Text Reading

In certain immersive experiences, users may scan their eyes back and forth across the same region repeatedly to understand the image content. Text reading forces this movement pattern, and thus is an effective way to measure how our sample preservation using WRS can be useful for users. To evaluate whether our sample accumulation using WRS can help improve reading speed under gaze-contingent foveation, we conducted a text-reading experiment similar to [Albert et al. 2019]. They found that gaze-contingent foveation can severely impact the reading speed of fast users, but does not have much of an effect on slower readers. However, their

conditions largely do not restrict the visual span of typical readers, and other works that focus on reducing visual span of readers find a decrease in reading speed when visual span is restricted [Chung et al. 1998; Legge et al. 1997, 2007]. We investigate whether our sample accumulation method using WRS can mitigate some negative reading speed reductions due to reduced visual span.

5.1.1 Experiment Setup. Fourteen subjects were recruited for the experiment. One was excluded due to poor performance on the reading comprehension task (< 50% correct), leaving 13 subjects (age 20-31, $M=24.69$, 11 male / 2 female). All participants had normal or corrected-to-normal vision. Nine subjects were native English speakers, with the remaining all fluent in English.

All stimuli (Figure 3b) were presented on a 55 inch Samsung LS03D with a 4K (3840x2160 pixels) spatial resolution and 120 Hz temporal resolution. Peak luminance was $248\text{cd}/\text{m}^2$. Participants’ head position was fixed using a chinrest 71.5 cm from the screen, covering an 80° total field of view. We used the Tobii Eye Tracker 5 running at 133 Hz to track participants’ gaze location. Rendering was performed using Unity3D 6 and an NVIDIA RTX 4070 Ti Super. All framerates were locked to 120 frames per second.

Users were presented passages from 3rd to 5th grade reading comprehension exams, ranging from 43 to 57 words. The stimulus text was presented in the center of the display and spanned $\pm 20^\circ$ horizontal field of view and 10° vertical field of view. The foveal regions showed 20 characters left-to-right at 10° , 15 at 7.5° , and 10 at 5° . Typical visual span for an English reader is around 15 characters rightward and 3-4 characters leftward [McConkie and Rayner 1975]. Peripheral nonsense text surrounded the stimuli text and encompassed the remaining peripheral vision. All text used the Times New Roman serif font. To severely limit the visual span, we used downsampling with bilinear filtering, rendering every mid-peripheral pixel (from the edge of the foveal region out to 15° diameter = 7.5° radius) at a shading rate of 8×8 pixels, and every far peripheral pixel at a shading rate of 16×16 pixels. A 1° linear blend was used between regions.

5.1.2 Procedure. To begin the experiment, participants were presented three trials to measure baseline reading speed. Participants

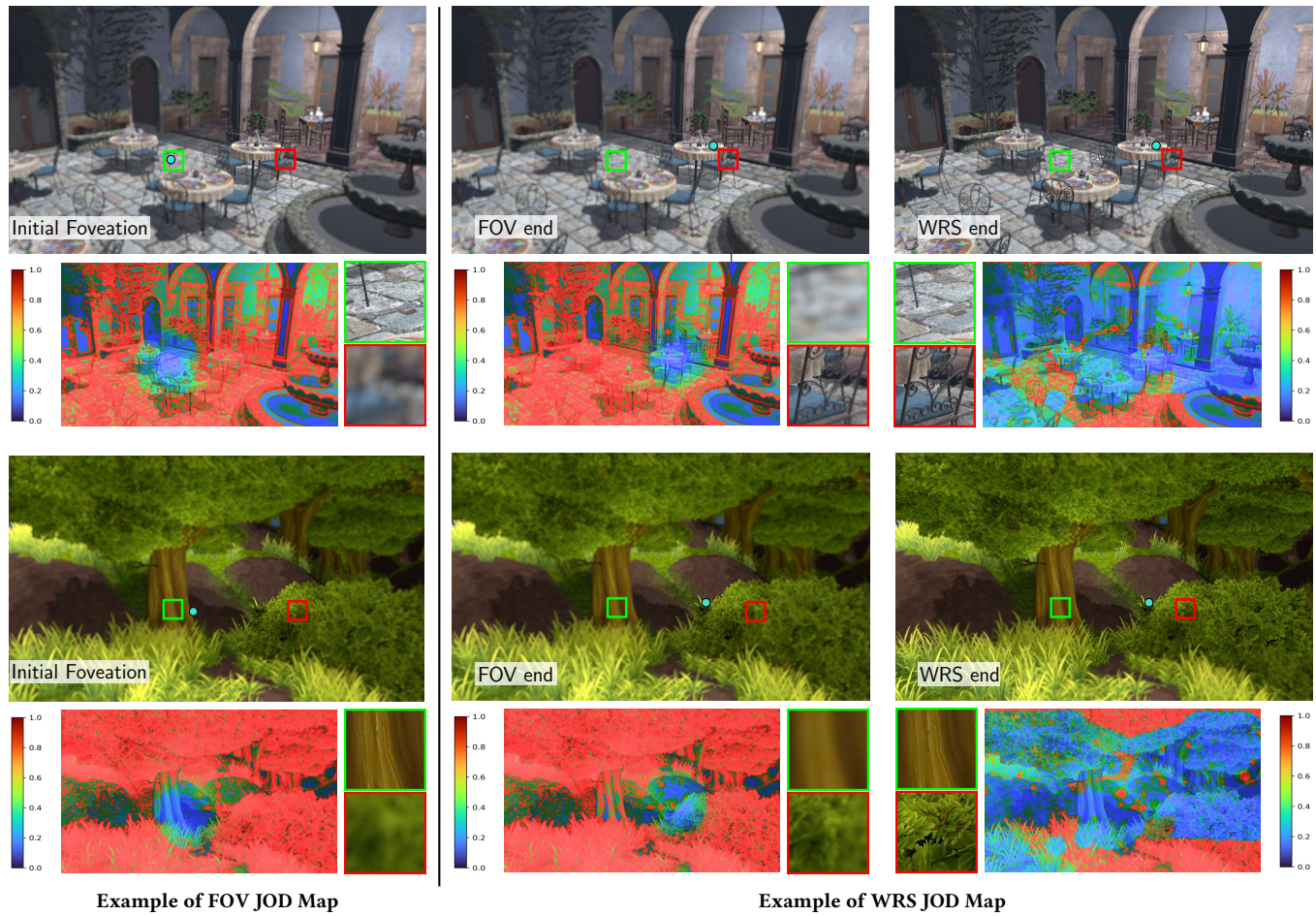


Figure 4: Images and FovVideoVDP JOD maps for FOV and WRS, both using a 5° fovea, and undergoing the same scanpath as in Figure 3a. Each image started from an initial foveation image on frame 0 (left), and progressed to their final renders (right). FovVideoVDP scores are detailed in Table 1.

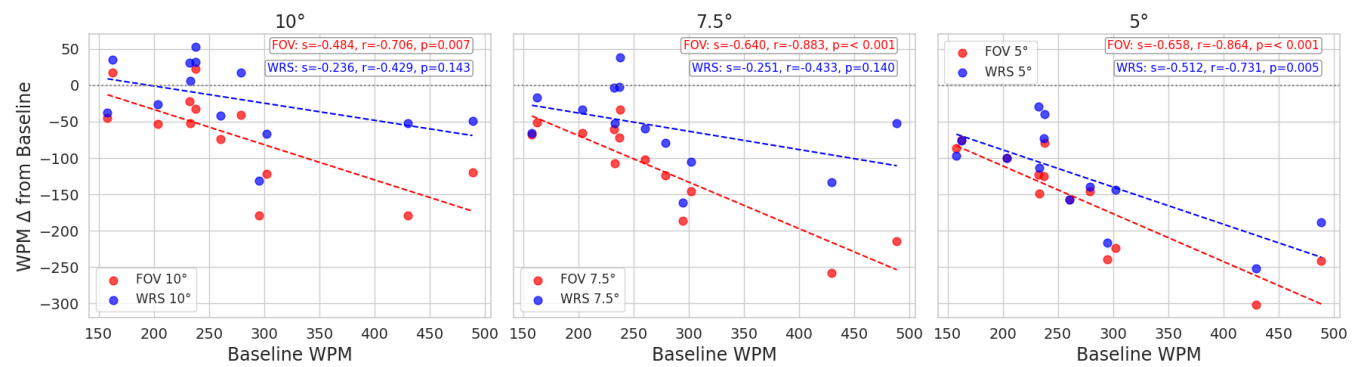


Figure 5: WPM differences from baseline across six rendering conditions. For each fovea size, blue shows the WPM decrease under WRS conditions, and red shows the WPM decrease under FOV conditions. Slopes (s), Pearson correlations (r), and p values for the decrease are shown. WRS 10° and WRS 7.5° did not significantly decrease reading speed, while all other conditions did; in all cases, WRS reduced reading speed significantly less than FOV.

were then presented with 6 conditions in random order with 5 unique text passages per condition (30 trials total): Foveated rendering (FOV) with foveal sizes of 5°, 7.5°, and 10°, and the same foveal sizes under WRS (WRS). Participants silently read the passages and pressed the keyboard when they were done. To assess their reading comprehension, they were presented 3 possible word choices that appeared in the paragraph they just read, and asked to select the correct word. One word appeared in the passage, and the other two were randomly chosen from unique words in other passages. They then rated how much they agree with the statement, "This text was easy to read under this rendering condition" on a Likert scale from 1 (Strongly Agree) to 7 (Strongly Disagree). Average accuracy was 77.46% (SD=8.03%), and no included participant was below 2 standard deviations from average.

5.1.3 Results. WPM differences from baseline were computed for each participant. Every condition (2 Rendering Condition x 3 Foveal Sizes) passed the Shapiro-Wilks test for normality. Mauchly's test indicated that the assumption of sphericity had been violated, ($W = 0.026$, $\chi^2(14) = 36.97$, $p = .001$). Degrees of freedom were corrected using the Greenhouse–Geisser estimate $\epsilon = 0.412$. We use a 2-way repeated-measures ANOVA with the rendering method (FOV vs WRS) and foveal size (5°, 7.5°, 10°) as within-subjects factors. There was a significant main effect of foveal size on WPM change, $F(2, 24) = 108.31$, $p < 0.001$, $\eta_G^2 = 0.302$, and a significant main effect of rendering method, $F(1, 12) = 31.91$, $p < 0.001$, $\eta_G^2 = 0.128$. There was a significant interaction between foveal size and rendering method, $F(2, 24) = 4.04$, $p = 0.031$, $\eta_G^2 = 0.008$.

We found a strong negative correlation between the methodology tested and reading speed in FOV 10° ($r = -.71$, $p = 0.007$), FOV 7.5° ($r = -.88$, $p < 0.001$), FOV 5° ($r = -.86$, $p < 0.001$), and WRS 5° ($r = -.73$, $p = 0.005$). We found moderate negative correlations that were not significant with WRS 10° ($r = -.43$, $p = .143$) and WRS 7.5° ($r = -.43$, $p = .140$). Full results are shown in Figure 5. While reading speed negatively deviated from baseline as a result of foveation in general across FOV and WRS methods, our results indicate that our temporal sample accumulation using WRS can mitigate some of the negative impacts on reading performance. Subjective preference scores are shown in Figure 3c.

5.2 Experiment 2: Perceptual Validation

WRS with sample accumulation was also tested on natural scenes with a moving camera, to ensure that the temporal accumulation under camera movement does not produce temporal artifacts from inaccurate reprojections. A 2-Interval Forced Choice (2IFC) experiment was run to investigate the utility of our reservoir sampling method (WRS) against standard foveated rendering (FOV) on stimuli that resemble common graphics use-cases more. The same three fovea sizes as used in Experiment 1 were used.

5.2.1 Experiment Setup. 12 participants (age 21–31, $M=26.25$; 5 female, 7 male) were recruited. All participants had normal or corrected-to-normal vision. A similar hardware setup as in 5.1 was used. The screen used was a 27 inch Dell S2725QS with a peak luminance of $262.5\text{cd}/\text{m}^2$. Participants were fixed in a chinrest 42.84 cm from the screen, giving a 70° total field of view. Because not all the environments used could natively render at 120 frames per second

at 4K spatial resolution, the framerate was locked to 60 frames per second. To keep experiment time low and prevent participant fatigue, participants saw a cascading set of comparisons for the 2IFC task: (FOV 10° vs WRS 10°), (FOV 10° vs WRS 7.5°), (FOV 10° vs WRS 5°), (FOV 7.5° vs WRS 7.5°), (FOV 7.5° vs WRS 5°), (FOV 5° vs WRS 5°). Users saw 3 scenes (Sponza, San Miguel, Fantasy Forest).

5.2.2 Procedure. Participants were presented a cross and asked to fixate on it for the entirety of the stimulus period. They were presented a moving camera trajectory (4s) that was the same for all conditions. They would see the first condition (4s), a 0.5s black screen, and then the same camera trajectory (4s) under the second condition. To maximize the pixel movement, exclusively lateral camera movement was used. Participants were asked which sequence was more clear. Because a 2IFC asks the participant to compare the stimulus they most recently saw to their memory of the first stimulus, stimuli were presented in random order such that one condition was shown first 3 times, and the comparison condition was also shown first 3 times. In total, users saw 6 comparisons 6 times, across 3 scenes, for 108 trials total. Trials were discarded and repeated at random if the user's gaze deviated more than 2.5° from the fixation point. Users were provided breaks between scenes, as well as whenever they desired. Total experimentation time took around 25 minutes.

5.2.3 Results. Since the order of presentation for the reservoir-based and foveated rendering methods was randomized, we conducted a one-sided binomial test for each condition pair to determine whether the observed preference for reservoir rendering exceeded chance. Each condition comprised 216 trials. Participants significantly preferred reservoir rendering over foveated rendering in the following comparisons: WRS 10° vs. FOV 10° (68% prefer WRS, $p < 0.001$), WRS 7.5° vs. FOV 10° (57% prefer WRS, $p < 0.01$), WRS 7.5° vs. FOV 7.5° (71% prefer WRS, $p < 0.001$), WRS 5° vs. FOV 7.5° (62% prefer WRS, $p < 0.001$), and WRS 5° vs. FOV 5° (64% prefer WRS, $p < 0.001$). Preference for WRS 5° vs. FOV 10° did not reach statistical significance (51% prefer FOV, $p = 0.076$), suggesting that participants could not differentiate between these two methods. These results show that our method provides better perceived image quality than foveated rendering during camera motion, and can achieve comparable perceptual quality even when rendering a smaller central region.

5.3 Performance

5.3.1 Image Quality. Perceptual image metrics for WRS vs FOV are presented in Table 1 for scenes rendered with a 5° foveal region and 15° mid-periphery. Our WRS method enables high levels of temporal accumulation, allowing for better perceptual image quality. All sequences were evaluated on PSNR, SSIM, LPIPS, and FovVideoVDP [Mantiuk et al. 2021]. A 5s scanpath taken from an eye tracker (Figure 3a) was used to evaluate these metrics. Images and FovVideoVDP heatmaps for the scanpaths are shown in Figure 4. In every case, WRS more closely matches the perceptual quality of the scene without foveation.

To evaluate dynamic camera motion, horizontal back-and-forth motion was used to maximize vection. We compare WRS to FOV +

TAA, and present those results in Table 1. We find that the temporal reuse of our method outperforms the image quality of TAA applied onto foveated rendering. In highly reflective scenes, some artifacts appear due to imprecise pixel reprojection with reflections; despite that, WRS reports higher FovVideoVDP scores than FOV + TAA.

Table 1: Per-scene perceptual image metrics of WRS and FOV. WRS-S and FOV-S use scanpaths generated in Figure 3a. WRS-M and FOV-M use lateral camera motion. FOV-M applies TAA. The scanpath was not used on Corridor.

Scene	Metric	WRS-S	FOV-S	WRS-M	FOV-M
Fantasy Forest	PSNR $\uparrow$ (dB)	23.71	21.93	24.61	23.39
	SSIM $\uparrow$	0.661	0.54	0.654	0.600
	LPIPS $\downarrow$	0.178	0.409	0.264	0.423
	FovVideoVDP $\uparrow$	7.350	6.386	7.717	6.676
Sponza	PSNR $\uparrow$ (dB)	26.52	24.13	26.52	24.10
	SSIM $\uparrow$	0.720	0.540	0.720	0.602
	LPIPS $\downarrow$	0.165	0.356	0.165	0.388
	FovVideoVDP $\uparrow$	7.892	7.118	7.892	7.167
San Miguel	PSNR $\uparrow$ (dB)	24.73	22.15	21.66	20.21
	SSIM $\uparrow$	0.749	0.579	0.693	0.614
	LPIPS $\downarrow$	0.138	0.325	0.231	0.391
	FovVideoVDP $\uparrow$	7.928	6.993	7.883	6.84
Corridor	PSNR $\uparrow$ (dB)	-	-	23.12	22.27
	SSIM $\uparrow$	-	-	0.7752	0.7359
	LPIPS $\downarrow$	-	-	0.2436	0.2833
	FovVideoVDP $\uparrow$	-	-	7.2186	7.0636

5.3.2 Runtime. Runtime performance is evaluated on a PC with an Intel i9-12900K CPU (3.2 GHz base), 32GB RAM, and an NVIDIA RTX 4070 Ti S GPU. We measure the final render performance, from initial rasterization, our foveated postprocess, and finally our WRS method running on top of foveated output. On all maps, our WRS method runs extremely quickly: On Sponza, San Miguel, and Fantasy Forest, all WRS conditions ran in under 0.1ms over foveated rendering while providing much higher perceptual image quality. On the highly reflective Corridor scene, our method added 0.94ms, but was quicker and higher fidelity than FOV + TAA.

Table 2: Runtimes (ms/frame) of Weighted Reservoir Sampling (WRS), Foveated Rendering (FOV), and Foveated Rendering with TAA (FOV + TAA).

MAP	WRS	FOV	FOV + TAA
Fantasy Forest	12.63	12.52	14.95
Sponza	9.82	9.77	11.97
San Miguel	13.59	13.53	18.62
Corridor	10.81	9.87	11.66

6 Limitations

Our results show positive improvements over traditional foveated rendering, and foveated rendering with other temporal reuse methods applied. However, our method is imperfect; when object occlusion or disocclusion occurs, we fall back to strict foveated rendering,

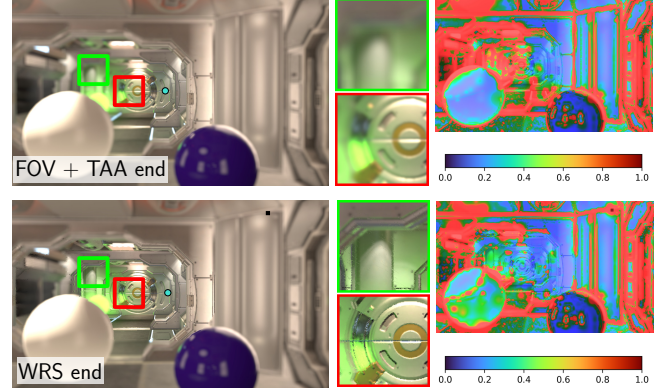


Figure 6: Highly reflective Corridor scene with dynamic objects and camera motion and rotation, rendered with FOV + TAA (top) and WRS (bottom). In rasterization-based reflective renders, simply using motion vectors can lead to inaccurate reprojection, causing artifacts to appear when temporally accumulating samples, as pixels get reprojected to incorrect locations, impacting the reflection quality. Despite this, WRS had higher FovVideoVDP scores.

forcing a reset of the temporal history of the reservoir. In scenes with many dynamic objects, this limits the efficacy of our implementation. Additionally, as our method works on foveated rasterization, reflections and refractions can be difficult to reproject properly, as the motion vectors are no longer accurate. This can lead to artifacting from incorrect reprojection, producing a speckled effect, seen on the highly reflective Corridor scene (Figure 6). Despite this, our WRS method produces higher FovVideoVDP quality over FOV + TAA, which is promising for future improvements.

7 Conclusion and Discussion

Foveated rendering allows for computational savings for high-resolution rendering tasks by rendering peripheral regions at a lower spatial resolution. Traditional methods, however, do not tend to temporally save previously high-resolution pixels between renders, potentially losing out on opportunities due to saccadic eye behaviour. In this paper, we introduced Weighted Reservoir Sampling to Foveated Rendering in rasterization contexts to temporally accumulate high-quality samples. Our key insight is that during fixation, saccades, and upon saccadic landing, smaller ocular drifts and microsaccades occur to enhance details in foveal vision. With traditional eye-tracked foveated rendering approaches, any samples that were foveal samples in a prior frame but not in the current one will be discarded and rendered at a lower quality resolution. However, by stochastically accumulating perceptually relevant frames and combining that with a new temporal bias scheme based on a Binomial distribution, we can achieve high levels of temporal accumulation at a very low computational cost.

References

Charu C Aggarwal. 2006. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd international conference on Very large data bases*. 607–618.

- Rachel Albert, Anjul Patney, David Luebke, and Joohwan Kim. 2017. Latency requirements for foveated rendering in virtual reality. *ACM Transactions on Applied Perception (TAP)* 14, 4 (2017), 1–13.
- Rachel A Albert, Angelica Godínez, and David Luebke. 2019. Reading speed decreases for fast readers under gaze-contingent rendering. In *ACM symposium on applied perception* 2019. 1–6.
- Maliha Ashraf, Rafal K Mantiuk, Alexandre Chapiro, and Sophie Wuerger. 2024. castleCSF—A contrast sensitivity function of color, area, spatiotemporal frequency, luminance and eccentricity. *Journal of vision* 24, 4 (2024), 5–5.
- H Aubert and CFR Foerster. 1857. Beiträge zur Kenntniss des indirecten Sehens.(I). Untersuchungen über den Raumsinn der Retina. *Archiv für Ophthalmologie* 3, 2 (1857), 1–37.
- Carl Bergmann. 1854. Zur Kenntniss des gelben Flecks der Netzhaut. *Zeitschrift für rationelle Medizin* 3 (1854), 245–52.
- Carl Bergmann. 1857. *Anatomisches und Physiologisches über die Netzhaut des Auges*. Verlag nicht ermittelbar.
- Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 148–1.
- Yancheng Cai, Ali Bozorgian, Maliha Ashraf, Robert Wanat, and K Rafal Mantiuk. 2024. elaTCSF: A Temporal Contrast Sensitivity Function for Flicker Detection and Modeling Variable Refresh Rate Flicker. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.
- Ville Cantory and Nathan Ringo. 2023. An image-space split-rendering approach to accelerate low-powered virtual reality. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 893–894.
- Min-Te Chao. 1982. A general purpose unequal probability sampling plan. *Biometrika* 69, 3 (1982), 653–656.
- Susana TL Chung, J Stephen Mansfield, and Gordon E Legge. 1998. Psychophysics of reading. XVIII. The effect of print size on reading speed in normal peripheral vision. *Vision research* 38, 19 (1998), 2949–2962.
- Christine A Curcio and Kimberly A Allen. 1990. Topography of ganglion cells in human retina. *Journal of comparative Neurology* 300, 1 (1990), 5–25.
- Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. 1990. Human photoreceptor topography. *Journal of comparative neurology* 292, 4 (1990), 497–523.
- Lamine Diop, Marc Plantevit, and Arnaud Soulet. 2024. RPS: A Generic Reservoir Patterns Sampler. In *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 110–115.
- Raymond Dodge. 1900. Visual perception during eye movement. *Psychological review* 7, 5 (1900), 454.
- Pavlos S Efraimidis. 2015. Weighted random sampling over data streams. *Algorithms, Probability, Networks, and Games: Scientific Papers and Essays Dedicated to Paul G. Spirakis on the Occasion of His 60th Birthday* (2015), 183–195.
- Pavlos S Efraimidis and Paul G Spirakis. 2006. Weighted random sampling with a reservoir. *Information processing letters* 97, 5 (2006), 181–185.
- Benno Erdmann and Raymond Dodge. 1988. *Psychologische Untersuchungen über das Lesen auf experimenteller Grundlage*. Niemeyer.
- CT Fan, Mervin E Muller, and Ivan Rezucha. 1962. Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *J. Amer. Statist. Assoc.* 57, 298 (1962), 387–402.
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D graphics. *ACM transactions on Graphics (TOG)* 31, 6 (2012), 1–10.
- Yong He, Yan Gu, and Kayvon Fatahalian. 2014. Extending the graphics pipeline with adaptive, multi-rate shading. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.
- Emile Javal. 1878. Essai sur la physiologie de la lecture. *Annales d’Ocillistique* 80 (1878), 97–117.
- Anton S Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. 2019. DeepFovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–13.
- Jonathan Korein and Norman Badler. 1983. Temporal anti-aliasing in computer generated animation. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*. 377–388.
- Philip Kortum and Wilson S Geisler. 1996. Implementation of a foveated image coding system for image bandwidth reduction. In *Human Vision and Electronic Imaging*, Vol. 2657. SPIE, 350–360.
- Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. 2021. A perceptual model for eccentricity-dependent spatio-temporal flicker fusion and its applications to foveated graphics. *ACM transactions on graphics (TOG)* 40, 4 (2021), 1–11.
- Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. 2023. Towards attention-aware foveated rendering. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–10.
- Yuna Kwak, Eric Penner, Xuan Wang, Mohammad R Saeedpour-Parizi, Olivier Mercier, Xiuyun Wu, Scott Murdison, and Phillip Guan. 2024. Saccade-Contingent Rendering. In *ACM SIGGRAPH 2024 Conference Papers*. 1–9.
- Matteo Paolo Lanaro, Hélène Perrier, David Coeurjolly, Victor Ostromoukhov, and Alessandro Rizzi. 2020. Blue-noise sampling for human retinal cone spatial distribution modeling. *Journal of Physics Communications* 4, 3 (2020), 035013.
- Gordon E Legge, Sonia J Ahn, Timothy S Klitz, and Andrew Luebker. 1997. Psychophysics of reading—XVI. The visual span in normal and low vision. *Vision Research* 37, 14 (1997), 1999–2010.
- Gordon E Legge, Sing-Hang Cheung, Deyue Yu, Susana TL Chung, Hye-Won Lee, and Daniel P Owens. 2007. The case for the visual span as a sensory bottleneck in reading. *Journal of vision* 7, 2 (2007), 9–9.
- JY Lettvin. 1976. On seeing sidelong. *The Sciences*, 16 (4), 10–20.
- Gábor Liktó and Carsten Dachsbacher. 2012. Decoupled deferred shading for hardware rasterization. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*. 143–150.
- Rafal K Mantiuk, Maliha Ashraf, and Alexandre Chapiro. 2022. stelaCSF: a unified model of contrast sensitivity as the function of spatio-temporal frequency, eccentricity, luminance and area. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.
- Rafal 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 Transactions on Graphics (TOG)* 40, 4 (2021), 1–19.
- Ricardo Marroquim, Martin Kraus, and Paulo Roma Cavalcanti. 2007. Efficient Point-Based Rendering Using Image Reconstruction.. In *PBG@ Eurographics*. 101–108.
- George W McConkie and Keith Rayner. 1975. The span of the effective stimulus during a fixation in reading. *Perception & Psychophysics* 17 (1975), 578–586.
- Morgan McGuire. 2017. Computer Graphics Archive. <https://casual-effects.com/data>
- Adriano Melligrana. 2024. Investigating Methods for Weighted Reservoir Sampling with Replacement. *arXiv preprint arXiv:2403.20256* (2024).
- Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. 2018. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–20.
- Joerg H Mueller, Thomas Neff, Philip Voglreiter, Markus Steinberger, and Dieter Schmalstieg. 2021. Temporally adaptive shading reuse for real-time rendering and virtual reality. *ACM Transactions on Graphics (TOG)* 40, 2 (2021), 1–14.
- Rafael Navarro, Pablo Artal, and David R Williams. 1993. Modulation transfer of the human eye as a function of retinal eccentricity. *Journal of the Optical Society of America A* 10, 2 (1993), 201–212.
- Diego Nehab, Pedro V Sander, Jason Lawrence, Natalya Tatarchuk, and John R Isidoro. 2007. Accelerating real-time shading with reverse reprojection caching. In *Graphics hardware*, Vol. 41. 61–62.
- Toshikazu Ohshima, Hiroyuki Yamamoto, and Hideyuki Tamura. 1996. Gaze-directed adaptive rendering for interacting with virtual space. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. IEEE, 103–110.
- Jorge Otero-Millan, Stephen L Macknik, Rachel E Langston, and Susana Martinez-Conde. 2013. An oculomotor continuum from exploration to fixation. *Proceedings of the National Academy of Sciences* 110, 15 (2013), 6175–6180.
- Yaobin Ouyang, Shiqiu Liu, Markus Kettunen, Matt Pharr, and Jacopo Pantaleoni. 2021. ReSTIR GI: Path resampling for real-time path tracing. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 17–29.
- Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–12.
- Andreas Polychronakis, George Alex Koulteris, and Katerina Mania. 2021. Emulating foveated path tracing. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 1–9.
- Whitman Richards. 1969. Saccadic suppression. *Journal of the Optical Society of America* 59, 5 (1969), 617–623.
- DA Robinson. 1964. The mechanics of human saccadic eye movement. *The Journal of physiology* 174, 2 (1964), 245.
- Michele Rucci, Ramon Iovin, Martina Poletti, and Fabrizio Santini. 2007. Miniature eye movements enhance fine spatial detail. *Nature* 447, 7146 (2007), 852–855.
- Michele Rucci and Martina Poletti. 2015. Control and functions of fixational eye movements. *Annual review of vision science* 1, 1 (2015), 499–518.
- Daniel Scherzer, Stefan Jeschke, and Michael Wimmer. 2007. Pixel-correct shadow maps with temporal reprojection and shadow test confidence. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 45–50.
- Daniel Scherzer, Lei Yang, Oliver Mattausch, Diego Nehab, Pedro V Sander, Michael Wimmer, and Elmar Eisemann. 2011. A survey on temporal coherence methods in real-time rendering. In *EUROGRAPHICS 2011 state of the art reports*. Eurographics Association, 101–126.
- Boubakar Sere, Christian Marendaz, and Jeanny Herault. 2000. Nonhomogeneous resolution of images of natural scenes. *Perception* 29, 12 (2000), 1403–1412.
- Michael Stengel, Steve Grogoric, Martin Eisemann, and Marcus Magnor. 2016. Adaptive image-space sampling for gaze-contingent real-time rendering. In *Computer graphics forum*, Vol. 35. Wiley Online Library, 129–139.
- Hans Strasburger, Jörg Huber, and David Rose. 2018. Ewald Hering’s (1899) on the limits of visual acuity: A translation and commentary: With a supplement on Alfred Volkmann’s (1863) physiological investigations in the field of optics. *i-Perception* 9,

- 3 (2018), 2041669518763675.
- Luca Surace, Cara Tursun, Ufuk Celikcan, and Piotr Didyk. 2023. Gaze-Contingent Perceptual Level of Detail Prediction. (2023).
- Nicholas T Swafford, José A Iglesias-Guitian, Charalampos Koniaris, Bochang Moon, Darren Cosker, and Kenny Mitchell. 2016. User, metric, and computational evaluation of foveated rendering methods. In *Proceedings of the ACM Symposium on Applied Perception*. 7–14.
- Taimoor Tariq and Piotr Didyk. 2024. Towards motion metamers for foveated rendering. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–10.
- Taimoor Tariq, Cara Tursun, and Piotr Didyk. 2022. Noise-based enhancement for foveated rendering. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.
- Larry N Thibos, David L Still, and Arthur Bradley. 1996. Characterization of spatial aliasing and contrast sensitivity in peripheral vision. *Vision research* 36, 2 (1996), 249–258.
- Larry N Thibos, DJ Walsh, and FE Cheney. 1987. Vision beyond the resolution limit: aliasing in the periphery. *Vision Research* 27, 12 (1987), 2193–2197.
- Norimichi Tsumura, Chizuko Endo, Hideaki Haneishi, and Yoichi Miyake. 1996. Image compression and decompression based on gazing area. In *Human Vision and Electronic Imaging*, Vol. 2657. SPIE, 361–367.
- Okan Tarhan Tursun, Elena Arabadzhiyska-Koleva, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. 2019. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- Karthik Vaidyanathan, Marco Salvi, Robert Toth, Theresa 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.
- Robert J Van Beers. 2007. The sources of variability in saccadic eye movements. *Journal of Neuroscience* 27, 33 (2007), 8757–8770.
- Stefan Van der Stigchel, Martijn J Schut, Jasper Fabius, and Nathan Van der Stoep. 2020. Transsaccadic perception is affected by saccade landing point deviations after saccadic adaptation. *Journal of vision* 20, 9 (2020), 8–8.
- Jeffrey Scott Vitter. 1984. Faster methods for random sampling. *Commun. ACM* 27, 7 (1984), 703–718.
- Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.
- Frances C Volkman. 1962. Vision during voluntary saccadic eye movements. *Journal of the Optical Society of America* 52, 5 (1962), 571–578.
- David R Walton, Rafael Kuffner Dos Anjos, Sebastian Friston, David Swapp, Kaan Akşit, Anthony Steed, and Tobias Ritschel. 2021. Beyond blur: Real-time ventral metamers for foveated rendering. *ACM Transactions on Graphics* 40, 4 (2021), 1–14.
- Lili Wang, Xuehuai Shi, and Yi Liu. 2023. Foveated rendering: A state-of-the-art survey. *Computational Visual Media* 9, 2 (2023), 195–228.
- Andrew B Watson. 2014. A formula for human retinal ganglion cell receptive field density as a function of visual field location. *Journal of vision* 14, 7 (2014), 15–15.
- Martin Weier, Thorsten Roth, Ernst Kruijff, André Hinkenjann, Arsène Pérard-Gayot, Philipp Slusallek, and Yongmin Li. 2016. Foveated real-time ray tracing for head-mounted displays. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 289–298.
- David R Williams, Pablo Artal, Rafael Navarro, Matthew J McMahon, and David H Brainard. 1996. Off-axis optical quality and retinal sampling in the human eye. *Vision research* 36, 8 (1996), 1103–1114.
- Chris Wyman, Markus Kettunen, Daqi Lin, Benedikt Bitterli, Cem Yuksel, Wojciech Jarosz, and Paweł Kozłowski. 2023. A gentle introduction to restir path reuse in real-time. In *ACM SIGGRAPH 2023 Courses*. 1–38.
- Chris Wyman and Alexey Panteleev. 2021. Rearchitcting spatiotemporal resampling for production. In *Proceedings of the Conference on High-Performance Graphics*. 23–41.
- Lei Yang, Shiqiu Liu, and Marco Salvi. 2020. A survey of temporal antialiasing techniques. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 607–621.
- 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.
- Bert L Zuber, Lawrence Stark, and G Cook. 1965. Microsaccades and the velocity-amplitude relationship for saccadic eye movements. *Science* 150, 3702 (1965), 1459–1460.