Computational Methods for Immersive Perception

A Dissertation presented

by

Qi Sun

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

May 2018

Stony Brook University

The Graduate School

Qi Sun

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Arie Kaufman - Dissertation Advisor
Distinguished Professor, Department of Computer Science

Hong Qin - Chairperson of Defense
Professor, Department of Computer Science

Xiaojun Bi
Assistant Professor, Department of Computer Science

David Luebke
Vice President, NVIDIA Corporation

Li-Yi Wei
Senior Research Scientist, Adobe Research

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Computational Methods for Immersive Perception

by

Qi Sun

Doctor of Philosophy

in

Computer Science

Stony Brook University

2018

Immersive content authoring and consumption are critical factors for the next generation of virtual reality (VR) and augmented reality (AR) platforms. They are consisted of real-time rendering and interface software, and display hardware. This research is to understand and more importantly leverage human perceptual factors, such as locomotion, accommodation and foveation, for fundamental geometry/imaging/rendering algorithms and practical applications, such as VR/AR mapping and computationally foveated displays. This work primarily focuses on answering two questions in immersive platform: "where we are" and "what we see".

For the first question, we present a VR redirected locomotion system. This is a technique that allows users to explore large virtual environments from small physical environments via real walking with head mounted displays. The main idea is to warp the virtual scene such that users perceive a different world thus being guided to avoid physical walls and obstacles.

At a lower-level, understanding how the eye perceives immersive stimuli is another major challenge for delivering comfortable VR/AR experience. Our eyes

have a complex set of different behaviors, such as accommodation, saccade, blink etc. A key problem causing immersive simulator sickness is the vergence-accommodation conflict. Although light field displays can support proper accommodation, its 4D rendering workload usually results in high latency, which jeopardizes the comfort. Our perceptual studies estimate human foveated accommodation factors. Based on the psychophysical experiments, we further analyze display-lens-retinal bandwidths and formulate a content-adaptive sampling model in the 4D ray space. We verify our method by building a prototype light field display that can render only $16\% - 30\%$ rays without compromising perceptual quality.

Finally, utilizing the discoveries from answering both questions, we present another novel VR redirected walking system. It significantly expands the allowable size difference between the virtual and physical environments by exploiting temporal blindness in human visual systems. We look specifically at saccades, which are rapid eye movements during which the visual signals are temporarily blocked but not perceivable due to brain interpolation. Thus, during saccades, we can change the virtual cameras inside the HMD to nudge users to walk in the desired directions without them noticing the camera manipulation.

Table of Contents

Contents

List of Figures

List of Tables

Acknowledgements

I would like to first express my sincere appreciation to my PhD advisor, Distinguished Professor Arie Kaufman. The luckiest part of being able to work with him is the freedom of conducting whatever research I like to explore without pressure and worries of failures. I also thank my all-time mentor, Dr. Li-Yi Wei, for his tireless advising and guidance (in not only research/career but life/personality as well) since I was in college.

It is also my greatest honor to be able to work with so many all-star collaborators during my PhD career. Thanks Dr. Dave Luebke for offering me two opportunities of exploring different NVIDIA Research offices and taking time for valuable discussions from his busy management workload; other NVIDIANs including Fu-Chung (Jonash) Huang and Anjul Patney for mentoring me so much on techniques, presentation and career choice; Joohwan Kim for introducing me vision science and high standard of human studies; Chris Wyman for teaching me useful writing skills; Kaan Aksit for showing me how to build optics hardware, and Omer "Visibly Cool" Shapira, Morgan McGuire, Aaron Lefohn for brainstorming and hosting. On the other hand, I would like to also thank Cynthia Lu, Paul Asente and Nathan Carr for collaborating and/or introducing me to the Adobe Research team.

Thanks Prof. Hong Qin for his selfless guidance and discussion since I started to take his computer graphics course; other lab and intern mates such as Zhixin Shu, Kishore Rathinavel, Ping Hu and Saad Nadeem for their valuable discussions and help on my projects. I would also like to thank external collaborator Piotr Didyk and committee member Xiaojun Bi for their kind efforts.

At last, I thank my family including (grand)parents, especially my grandfather who always supports but left me during my PhD study. Thank Suwen Zhu for her selfless support of both my research and life!

Publications

1. Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, Arie Kaufman. Towards Virtual Reality Infinite Walking: Dynamic Saccadic Redirection. *ACM Trans. Graph.* 37, 4 (August 2018, to appear)
2. Qi Sun, Fu-Chung Huang, Joohwan Kim, Li-Yi Wei, David Luebke, Arie Kaufman. Perceptually-Guided Foveation for Light Field Displays. *ACM Trans. Graph.* 36, 6, Article 192 (November 2017)
3. Joohwan Kim, Qi Sun, Fu-Chung Huang, Li-Yi Wei, David Luebke, Arie Kaufman. Perceptual Studies for Foveated Light Field Displays. *arXiv preprint* arXiv:1708.06034, 2017.
4. Qi Sun, Li-Yi Wei, Arie Kaufman. Mapping Virtual and Physical Reality. *ACM Trans. Graph.* 35, 4, Article 64 (July 2016)
5. Qi Sun, Seyedkoosha Mirhosseini, Ievgeniia Gutenko, Ji Hwan Park, Charilaos Papadopoulos, Bireswar Laha, Arie Kaufman. Poster: Buyers Satisfaction in A Virtual Fitting Room Scenario Based on Realism of Avatar. *IEEE Symposium on 3D User Interfaces*, 2015
6. Koosha Mirhosseini, Qi Sun, Krishna Chaitanya Gurijala, Bireswar Laha, Arie Kaufman. Benefits of 3D Immersion for Virtual Colonoscopy. *IEEE Visualization Workshop on 3DVis* 2014

1 Overview

The growth of VR/AR industry leads to extensive demands of high quality content, fast speed rendering, comfortable and friendly user interface. Computational immersive perception considers the eye, the nervous system and the brain. The concept is presented to understand and leverage how human perceive a virtual scene from immersive VR/AR display devices.

At a higher level, natural navigation system can reduce sickness and help efficiency of immersive application. In such scenarios, users' positional and directional behaviors and feedbacks are critical for VR/AR locomotion.

On the other hand, our low level vision mechanism, for example, accommodation and foveation, is used to understand the natural world's appearances. However, current immersive displays lack enough cues to fully simulate a natural worlds, which is a major cause of simulator sickness. Leveraging the eye's accommodative and foveated features can address both rendering performance and visual discomfort.

Locomotion Ideally, a VR environment should facilitate full immersion and natural movement. Current devices such as projected rooms (CAVEs [1]) and head mounted displays (HMDs) can provide realistic rendering, but often require users to remain stationary or walk within a limited area due to hardware limitations (e.g., space or cables) or navigation concerns (e.g., real environments not visible through HMD). Users thus need to employ less natural means such as gamepads and walk-in-place devices to control their movements, which can negatively impact their sense of presence compared with natural interaction and real walking [5].

We present [6] a novel perspective to maximize presence in a given virtual world and facilitate real walking in a given physical world. With this technique, users can freely explore large virtual environments from small physical environments via real walking.

Our novel perspective to address this problem by distorting the visual stimuli, that is the rendered virtual scene through HMDs. Intuitively, in natural world, a road can either be straight or curved. The method consists of two key components: a planar map between virtual and real world floor plans, and a camera projection derived from the planar map and scene content. The former is globally surjective but

locally injective between the virtual and physical floor plans to minimize angular and distal distortions for walkthroughs. The latter preserves the virtual world appearance while observes the physical world geometry to balance between visual fidelity and navigation comfort.

Accommodation In immersive media, the depth cue is a key to visual comfort and interaction accuracy/efficiency thus better experience. Other than binocular cues (e.g., disparity, convergence) which are provided by current immersive displays, monocular cues, especially accommodation, is still missing in traditional display systems. As a result, vergence-accommodation conflict causes simulator sickness and confusing position response in interactive scenarios such as painting, design and gaming.

We measure, for the first time, the blur detection/discrimination and light field depth perception thresholds in up to 15 degree of visual eccentricity [7]. This low-level vision study rejects the idea of replacing the peripheral rendering with 2D billboards when sampling accommodation-supportive rays, that is, 4D light fields are still required. The psychophysical data motivates our further perceptual light field rendering as below and can also guide other foveated rendering approaches.

Foveation Although light field displays can support proper focal cue by 4D light rays thus solving vergence-accommodation conflict [8], they are considerably more costly to render or acquire than 2D images. That means they often lack sufficient speed or resolution for fully immersive VR applications which are sensitive to simulator sickness.

The human visual system can resolve higher spatial frequencies in the fovea than in the periphery. This property has been harnessed by recent 2D foveated rendering methods to reduce computation cost while maintaining perceptual quality [9].

Our work on foveated light field displays is inspired by this and our psychophysical studies [7]. We present foveated 4D light fields by investigating their effects on 3D depth perception [10]. Based on our psychophysical experiments and theoretical analysis on visual and display bandwidths, we formulate a content-adaptive importance model in the 4D ray space. We verify our method by building a prototype light ield display that can render only $16\% - 30\%$ rays without compromising perceptual quality.

2 Introduction

2.1 Applications of immersive media

Gaming and artistic designs are typical examples of VR/AR applications. Aside from those, there are also other VR/AR interfaces and applications such as immersive virtual colonoscopy navigation [1], (See Figure 1) and scanning-based real object reconstruction [11].

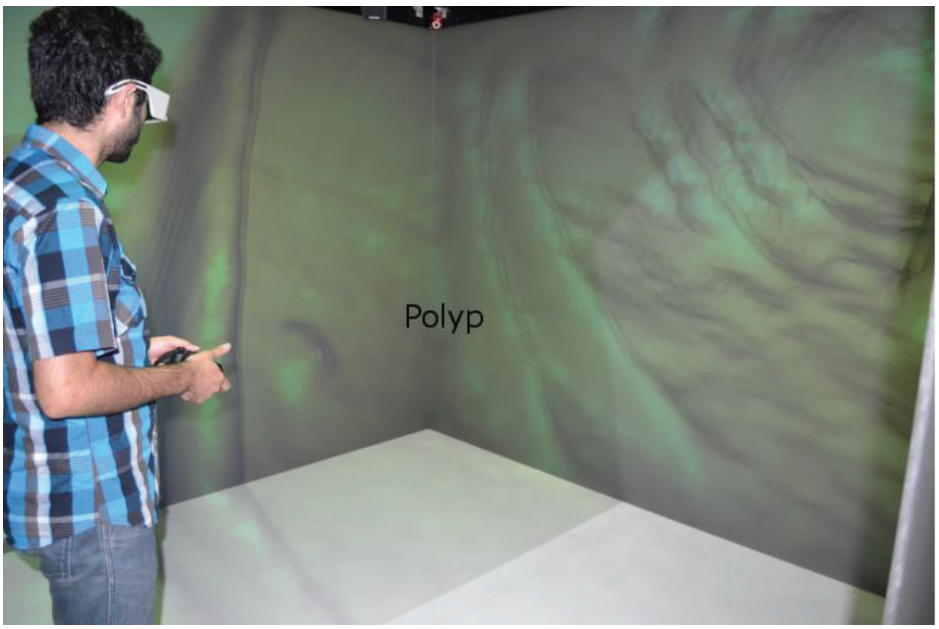


Figure 1: Immersive medical navigation from [1].

VR/AR environments can provide stronger immersion thus higher diagnose efficiency and accuracy for medical imaging. Virtual Colonoscopy (VC) [12] is a non-invasive clinical procedure that detects colon cancer in humans. VC seeks to supplement and improve the compliance rates for diagnosed patients, since the traditional optical colonoscopy is more unfriendly, and less effective for cancer detection. We discuss [1] the benefits of using a 3D immersive user interface for VC and various design choices for leveraging the effects of the various combinations of VR system components, as in previous VR empirical studies.

On the other hand, with the rapid development and wide-spread availability of

handheld market-level 3D scanner, character modeling has recently gained focus in both academia and industry. Virtual shopping applications have been widely-used in e-business. By porting the virtual shopping experience to immersive virtual reality systems, designers are able to make users feel present in the virtual environment, which is the subjective feeling of experiencing part of the virtual world. We present our parameter-based human avatar generation system and ongoing work on expanding the virtual shopping to the immersive virtual reality platforms employing natural user interfaces [11]. We discuss ideas to evaluate buyer satisfaction using our system.

2.2 Computationally Immersive Perception

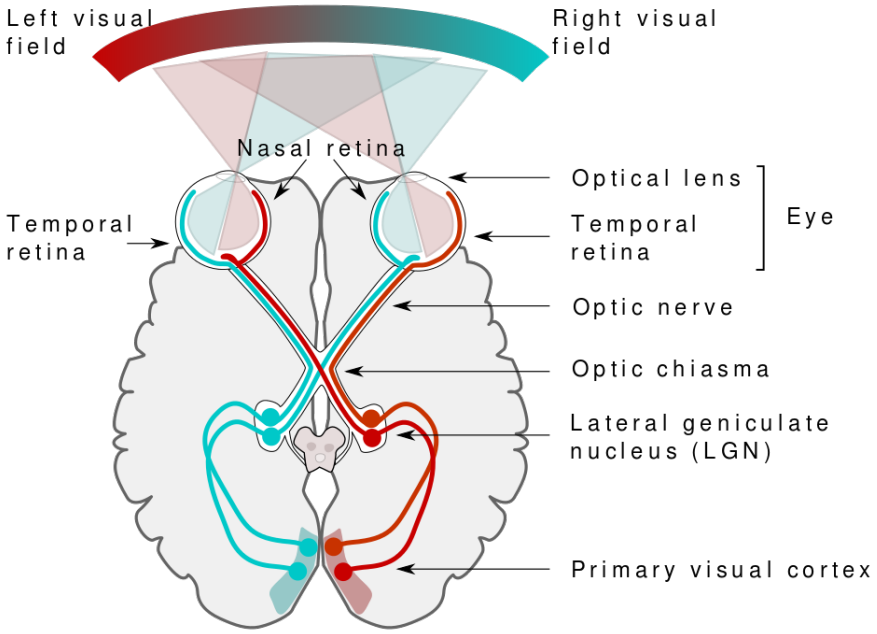


Figure 2: A simplified schema of the human visual pathway. *Image credit: Miquel Perello Nieto, courtesy of WikiCommons, Creative Commons 4.0 license*

The human vision is a complicated system involving the eye, the brain and the neural connection between them, as seen in Section 2.2. One critical challenge is

to understand and leverage human perceptual factors, such as locomotion, accommodation, and foveation.

Computational perception is to bridge high/low level perception and fundamental geometry/imaging/rendering algorithms. It can benefit immersive media authorization and consumption, such as VR/AR mapping and foveated displays.

2.3 Locomotion

(a) *large virtual scene* **(b)** *small physical space*

Figure 3: Static mapping examples. *(a) and (b) show a pair of large virtual and small physical room example. The size different limits the natural locomotion in VR.*

With the confluence of VR hardware and software developments, a variety of devices and setups are now offering different costs, features, and capabilities. Ideally, a VR environment should facilitate full immersion and natural movement. Current devices such as projected rooms (CAVEs) and head mounted displays (HMDs) can provide realistic rendering, but often require users to remain stationary or walk within a limited area due to hardware limitations (e.g., CAVE space or HMD cables) or navigation concerns (e.g., real environments not visible through HMD). Users thus need to employ less natural means such as gamepads and walk-in-place devices to control their movements, which can negatively impact their sense of presence compared with natural interaction [13] and real walking [5].

Combining realistic VR displays and real walking has the potential for immersive presence and natural interaction. Free walking is already viable on the hardware side, as upcoming HMDs are equipped with low cost motion trackers (e.g., $799

for HTC Vive including motion trackers compared to $52000 for OptiTrack motion capture). However, virtual and real worlds often differ significantly in sizes, shapes, and objects. Thus, proper mapping between the two is needed to offer a believable presence in the virtual world and feasible navigation in the real world (e.g., users remain perceptually comfortable and without bumping into objects). Devising such mapping remains an important open problem in VR display and navigation.

Techniques such as redirected locomotion [14, 15], distorted space [16, 17], and physical props [18] have shown promise for bridging the gap between virtual and real scenes. Behavioral studies, such as [19–21], have indicated the possibility of navigating a large-scale virtual environment while physically remaining in a reasonably small real space. However, those existing methods use procedurally generated content for feasibility studies, but do not provide general methods to map between a given pair of virtual and real environments. Since the virtual environment (e.g., a game or an architectural design) is usually orthogonal to the physical environment (which varies depending on the end users), a general method to bridge the two is crucial for real VR walkthrough.

We propose a VR method to maximize presence in a given virtual world and facilitate real walking in a given physical world. Within the scope of this work we represent both worlds as planar floor plans, and use an HMD (Oculus DK2) attached to a notebook computer as the VR device to allow free navigation. Our method faithfully renders the virtual world inside the HMD but alters the camera projection for navigating the real environment, so that users can retain perceptual comfort while being guided to stay within the boundary walls and away from obstacles such as furniture. Figure 5 is an overview of our system and the user experience.

Our method consists of two key components: a planar map between virtual and real world floor plans, and a camera projection derived from the planar map and scene content. The planar map aims to preserve both angle and distance between the virtual and real worlds for visual and locomotive consistence. The camera rendering aims to preserve the virtual world appearance and the real world geometry, while guiding user navigation to avoid physical obstacles and vestibular discomfort.

Both planar maps [22–24] and projective rendering [25–27] have been extensively studied in computer graphics. However, our VR method has different requirements from these prior art. Prior planar maps often require bijectivity to avoid

folds, but our method does not require bijectivity as it looks for proper folding of the virtual world into the real one. We instead optimize for maps that can preserve angle and distance and can be efficiently computed for VR navigation. Prior projective rendering methods focus on speed and realism. In addition to that, our method also relies on the projection to properly guide user locomotion and hide perceptual discrepancies between the virtual and real worlds. We thus derive our camera projection according to the planar map and scene content to balance between visual realism, geometric consistency, and perceptual comfort.

Our method allows users to wear wireless HMDs to navigate virtual scenes via free locomotion in real worlds. We evaluate our system through a formative user study and applications in gaming, architecture walkthrough, and medical imaging.

In summary, the main contributions of this work include:

- An HMD-VR system that allows real walking in a given physical environment while perceiving a given virtual environment;
- A custom planar map that is globally surjective but locally injective between the virtual and physical floorplans to minimize angular and distal distortions for walkthroughs;
- Optimization methods to compute the aforementioned planar maps as two parts: a static forward map that minimizes angular and distal distortions while avoiding obstacles and boundaries, and a dynamic inverse map that guides natural locomotion and resolves local ambiguities;
- A rendering method that preserves the virtual world appearance while observes the physical world geometry to balance between visual fidelity and navigation comfort.

2.4 Accommodation and Foveation

Advances in graphics algorithms and hardware have enabled high quality and low latency for traditional 2D displays. However, consistent 3D depth perception, which is important for perceptual comfort, remains out of reach for many users.

Light field displays support focal cues [8,29–32], but current rendering techniques cannot generate high quality content in real time. With gaze tracking, foveated rendering reduces computational costs while maintaining perceptual quality [9,

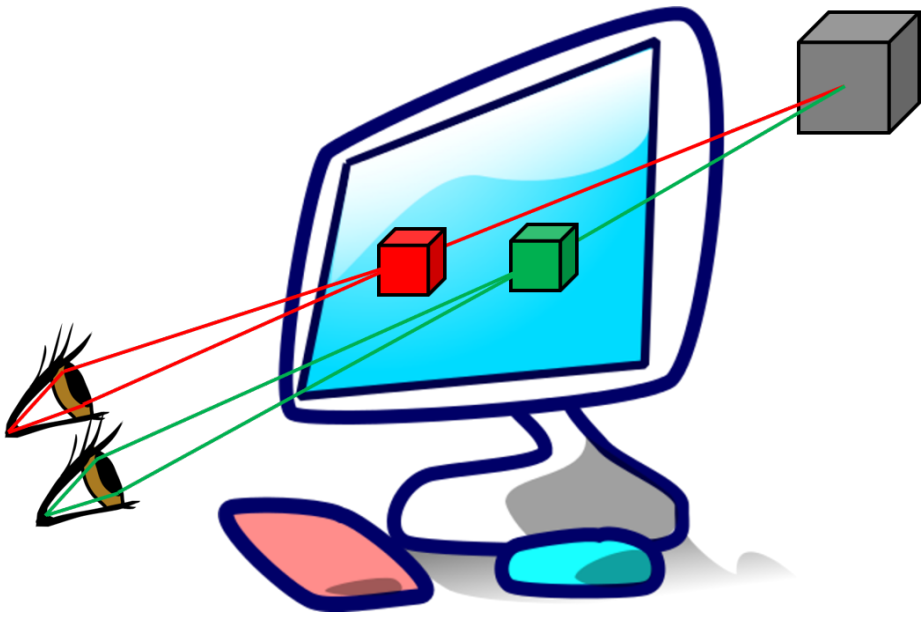


Figure 4: Illustration of vergence accommodation conflict. *In traditional stereo displays, the pupils move towards the estimated depth during focusing. However, the display cannot provide accommodative depth cues for the eye lens to defocus. This conflict is one of the major causes of simulator sickness in VR/AR [28]. Image credit: Li-Yi Wei.*

33]. However, existing methods are designed for 2D images; foveating 4D light field displays remains a challenging open problem. The human visual system automatically reconstructs 2D retinal images from 4D light fields. However, light field foveation cannot be simply reduced to image foveation due to the lack of reliable technology for tracking accommodation, a major factor of monocular depth perception.

Inspired by prior work on 4D light field display and 2D foveated image rendering, we present the first foveated light field rendering and display system that supports low latency and high quality, as well as focus accommodation to improve depth perception and reduce vergence-accommodation conflicts (illustrated in Section 2.4). Based on our psychophysical studies, our main idea is to derive an importance sampling model in the 4D light field ray space based on both foveation and accommodation. Conceptually, this can be achieved by tracing rays from retina cells back through the eye and into the scene, and varying the focal

length of the eye to sweep the ray space.

We derive the spectral bounds of the light field imaging pipeline, including the display, the eye lens, and the retina. Based on these bandwidths, we propose a sampling and reconstruction method for real-time rendering of foveated 4D light fields.

Our study also addresses a long-standing argument among the display and vision communities [8, 29, 32, 34–37] on the number of rays necessary to support focal cues. Our spectral analysis shows that the number depends on several factors including the display/eye optics, the retinal eccentricity, and the scene content. The analysis allows us to significantly reduce the rendering cost while preserving perceptual quality.

We evaluate our method by conducting psychophysical studies through our hardware prototype running a variety of scenes with different characteristics. Our system is shown to render up to $3\times$ faster than prior work and trace only $16\% \sim 30\%$ of all rays of the light field display while maintaining similar visual quality.

The main contributions of this work include:

- We analyze the bandwidth bounds for perceiving 4D light fields based on the display property, the eye lens, and the retinal distribution, and derive a minimum sampling rate to answer the argument among the display, graphics, and vision communities.
- Based on the spectral bounds and the depth perception measurements, we propose a 4D light field rendering method with importance sampling and a sparse reconstruction scheme, with reduced computation cost. The minimum 4D rendering supports both foveation and accommodation.
- We have built a hardware prototype for foveated light field display from commodity components including a gaze tracker, and a GPU-based light field rendering engine that runs in real time . Our prototype hardware + software system achieves better performance and quality than alternative methods, as verified through different scenes and user studies with multiple participants.

2.5 Saccadic Locomotion

Room-scale VR increases presence and decreases discomfort caused by visual-vestibular inconsistency by allowing the user to walk freely in a physical space [5]. However, a direct one-to-one mapping from virtual to physical space is impractical for most applications. Today's room-scale experiences either constrain the virtual space through scenario design or frequently interrupt the user and break their presence by requiring them to walk back to the center of the physical room or consciously teleport in the virtual world. A major challenge for VR is embedding a large virtual space within a small, irregular, multi-user physical space while minimizing interruptions. The ideal solution would create the perception of infinite walking in the virtual space within a small, finite physical space.

Treadmills or other physical devices can address the infinite walking problem, but are undesirable for general applications because they are expensive, bulky, and may compromise the user's balance, while also preventing free user movements such as kneeling and jumping. The current state of the art techniques for solving the mapping problem using only a head-mounted display (HMD) are redirected walking [14, 38, 39] and warping [6, 40]. These methods create a distorted mapping of the virtual environment by applying to the world subtle rigid-body and nonlinear transformations, respectively. These magnify the effective physical space, but state-of-the-art methods still require an unoccluded space of $36\,\mathrm{m}^2$ to be simultaneously imperceptible and effective [41]. This is a significant step towards practical room-scale VR for unconstrained scenarios, but it is still too large to accommodate many home and office rooms. We believe the main cause is the perceptually-imposed limitation of traditional redirection systems which cannot respond to the real-time user and environmental changes.

We present a novel, *dynamic* solution to the infinite walking problem. It is the first to be demonstrated as effective for physical areas as small as $12.25\,\mathrm{m}^2$. This significant advance beyond previous results meets for the first time the standard for practicality: these bounds match the recommended consumer HMD room-scale installation bounds, e.g., for HTC Vive and Oculus Rift. Our key innovation is redirecting the user much more aggressively, yet still imperceptibly, by tracking rapid eye movements called saccades using a HMD equipped with internal gaze-tracking cameras, and incorporating guided navigation and planning based on the scenario.

Saccades are rapid eye movements during which viewers are momentarily blind

in a phenomenon called *saccadic suppression*. Saccades occur frequently, but our high-level visual system prevents conscious awareness of the blindness. The visual system also essentially recalibrates its orientation after a saccade on the assumption that the world itself has not changed [42]. We exploit that assumption to change the virtual world imperceptibly and avoid predicted future collisions with physical objects. Our method retains faithful visual and vestibular experiences across a broader range of virtual and physical spaces than previous methods. To further enhance the effectiveness of the technique, we also employ subtle gaze directions to opportunistically trigger additional saccades and a content-aware path planner to adapt to dynamic environmental changes. Our main contributions are:

- The end-to-end redirected walking system based on saccadic suppression, effective for consumer room-scale VR;
- A real-time path planning algorithm which automatically avoids static and dynamic obstacles by responding to individuals' eye movements – our optimization links user behavior and physical changes, considers possibilities of near future through real-time sampling, and finds the best numerical solution for online camera manipulation;
- The use of subtle gaze direction (SGD) methods in VR to induce more saccades for the system to exploit;
- Validation through simulations and real redirected walking scenarios with game-like tasks, such as search and retrieval.

3 Locomotion

Real walking offers higher immersive presence for virtual reality (VR) applications than alternative locomotive means such as walking-in-place and external control gadgets, but needs to take into consideration different room sizes, wall shapes, and surrounding objects in the virtual and real worlds, as seen in Figure 3. Despite perceptual study of impossible spaces and redirected walking, there are no general methods to match a given pair of virtual and real scenes.

We propose a system to match a given pair of virtual and physical worlds for immersive VR navigation. We first compute a planar map between the virtual and physical floor plans that minimizes angular and distal distortions while conforming to the virtual environment goals and physical environment constraints. Our key idea is to design maps that are globally surjective to allow proper folding of large virtual scenes into smaller real scenes but locally injective to avoid locomotion ambiguity and intersecting virtual objects. From these maps we derive altered rendering to guide user navigation within the physical environment while retaining visual fidelity to the virtual environment. Our key idea is to properly warp the virtual world appearance into real world geometry with sufficient quality and performance. We evaluate our method through a formative user study, and demonstrate applications in gaming, architecture walkthrough, and medical imaging.

3.1 Previous Work

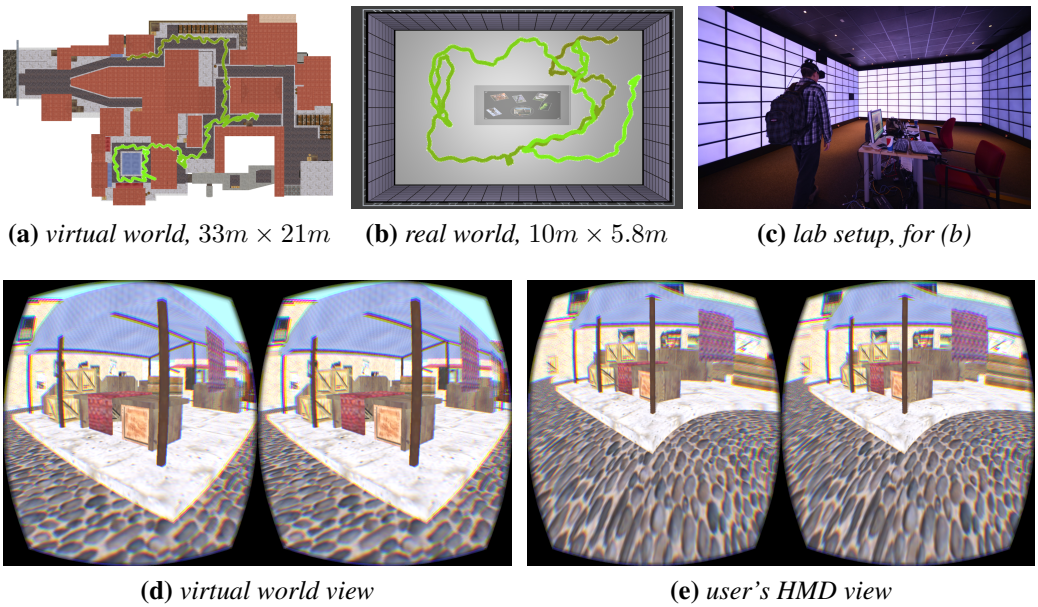
Immersive virtual environments There are various forms of immersive virtual environments. Some, such as rooms or cabins (CAVEs), offer semi-immersive experiences in which users can see virtual worlds projected on physical displays. Others, such as head mounted displays (HMDs), have more compact setup and fuller immersion than CAVEs, and have gained recent popularity due to improvement in hardware and software (see e.g., [8, 43]). However, users' immersive experiences depend on not only rendering/display performance but also interaction and navigation capabilities. HMDs can block perception of the surrounding real world with negative impacts on user interaction and navigation, such as hand motion [44], obstacle avoidance, and walking direction. Walking-in-place (WIP) techniques, such as omni-directional treadmills [45], robot tiles [46], and motion

carpets [47], can reduce some of these issues, but have yet to gain popularity due to barriers in hardware and usability (see e.g., [48]).

Real walking Studies have shown that real walking outperforms walking-in-place and other indirect means of VR navigation [5]. However, real walking requires sufficiently large physical spaces, which are almost always in different (usually smaller) sizes and shapes from the corresponding virtual spaces (unless the latter are designed from the former, as in [49]). Techniques such as physical props [18], warped spaces [16, 17], and redirected walking [38, 50–52] have been proposed to reconcile the virtual and physical worlds, and behavior studies have indicated that limited amounts of space distortion can be acceptable for VR navigation [19–21, 53]. However, existing methods are not general enough to map between a given pair of virtual and physical environments. Our work aims to address this important problem.

Planar mapping Various planar mapping methods have been proposed to achieve application-specific goals such as minimizing distortion and avoiding folding (see e.g., [22–24] and the references therein). Our system also relies on planar mapping, but has needs beyond existing methods. For example, sparse constrained deformation [54] and convex SOCP optimization [22] are not suitable for our problem, and we need local isometry for geodesics rather than global isometry for distance-preserving. Moreover, most traditional planar mapping and deformation applications are based on user manipulation. In our application, the output domain is pre-defined by the real space. We thus propose a custom planar mapping with application-specific objectives, constraints, and solvers.

Re-projective rendering Re-projective rendering has a long history in computer graphics, including image-based rendering [25], general camera models [27], and shading reuse for acceleration [26]. Our system also uses re-projective rendering for HMDs, but faces a unique challenge: combining the appearance of the virtual world and the geometry of the physical world to strike the right balance between visual fidelity and navigation comfort. We thus propose a custom re-projective rendering method to address this challenge.



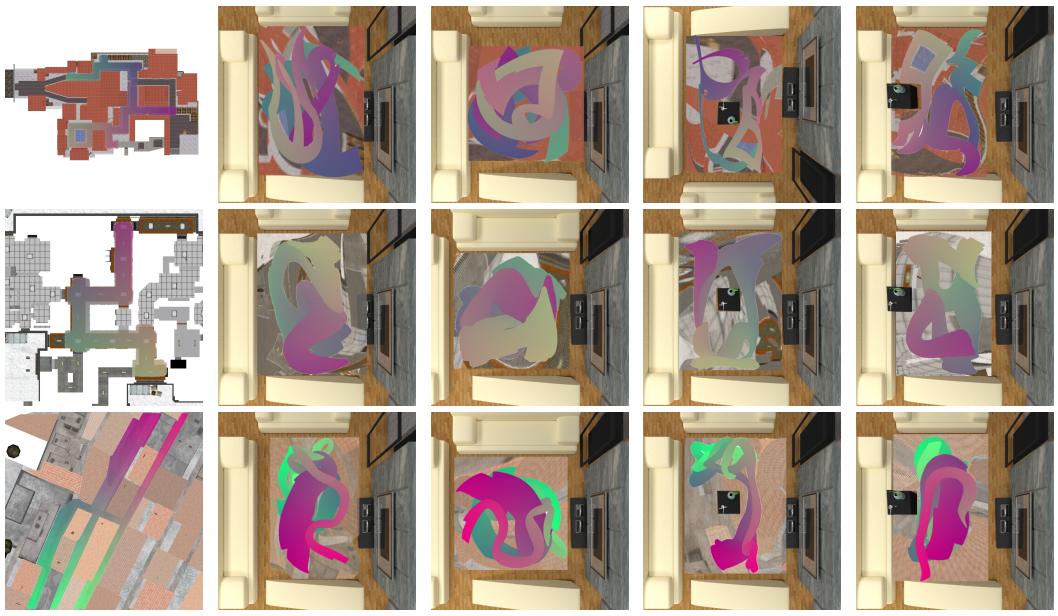
(a) *virtual world,* $33m \times 21m$ **(b)** *real world,* $10m \times 5.8m$ **(c)** *lab setup, for (b)*

(d) *virtual world view* **(e)** *user's HMD view*

Figure 5: Overview of our system. *(a) and (b) are aerial views of the virtual and physical worlds, with their dimensions in meters. Typically, (a) is much larger than (b). The goal is to enable the user to walk freely in (b) while experiencing (a) using a head-mounted display (HMD). Our system first computes a planar map between (a) and (b), with the mapped user walking paths overlaid in green gradients. (c) is a photograph of the lab setup, including the equipment and surroundings. Our system then renders the virtual world appearance (d) for the user's HMD view (e) in a way that is compatible with the real world geometry, so that the user can faithfully see the former and comfortably navigate the latter. Note that even though the user's view is entirely blocked by the HMD, our system guides the user away from boundaries and obstacles such as the walls and furniture. (Scene in (a) is courtesy of Counter Strike, Italy.)*

3.2 Method

Given the 2D floor plans for the virtual S_v and real S_r scenes, we first compute a static forward map $\mathbf{f}$ from S_v to S_r (Section 3.2.1). This map is surjective but not bijective in general when $S_v > S_r$, but should minimize both distance and angle distortion for VR walkthroughs. It should reach every point in both S_v and S_r, while keeping inside S_r and away from interior obstacles. Folding is introduced



(a) 200×200 *virtual* **(b)** 60 × 100 *real* **(c)** 70 × 70 *real* **(d)** 60 × 100 *real* **(e)** 60 × 100 *real*

Figure 6: Static mapping examples. *(a): original input virtual scene overlaid with the user paths. (b) and (c): map to real spaces with different sizes and shapes. (d) and (e): map to real spaces with interior obstacles within and adjacent to the boundaries. (Scenes courtesy from top to bottom: Counter Strike Italy, Counter Strike office, and Venice.)*

without tearing or breaking apart the virtual world.

At run time during user navigation, we compute a dynamic reverse map of $\mathbf{f}$ to determine the virtual location in S_v from the tracked user position in S_r (Section 3.2.2). This reverse map should be consistent with the forward map $\mathbf{f}$ while maintaining motion and perception consistency for the users.

Finally, we render the virtual scene into the HMD (Section 3.2.3). The rendering should have enough quality and speed, and fit the appearance of the virtual scene into the geometry of the real scene to balance between visual and motion fidelity.

3.2.1 Static Forward Mapping

In this step, we surjectivly map each virtual scene pixel $\mathbf{x} = (x, y) \in S_v$ to a real scene point $\mathbf{u} = (u, v) \in S_r$, where S_v and S_r represent 2D planar regions. Unlike most prior 2D planar mapping methods, our application does not require global bijectivity to allow proper folding of large virtual scenes into small real scenes. Instead, our map relies more on conformality and isometry to minimize angular and distal distortion during VR navigation. Figure 6 shows mapping examples with different inputs and outputs.

Inputs and outputs Both the virtual S_v and real S_r scenes are represented by planar 2D regions bounded by their external and internal boundaries (for domain shapes and obstacle regions, respectively). For computational purposes, we represent both spaces as polygonal shapes. In practice, those polygons can be extracted as convex/non-convex hulls from scanned data or design figures.

Representation Similar to prior meshless warping and planar mapping methods [22, 24], our method also adopts a basis-function form to facilitate analytical computation of Jacobians and Hessians:

$$(u(x, y),\ v(x, y)) = \mathbf{u} = \mathbf{f}(\mathbf{x}) = \sum_{i=1}^{p} \mathbf{c}_i \mathbf{b}_i(\mathbf{x}) + \mathbf{T}\mathbf{x}, \tag{1}$$

where $\{\mathbf{b}_i\}$ are basis functions with weights $\{\mathbf{c}_i\}$, and $\mathbf{T}$ is an affine transformation matrix. We use Gaussians for $\mathbf{b}$, that is,

$$\mathbf{b}_i(\mathbf{x}) = e^{\frac{-|\mathbf{x}-\mathbf{x}_i|^2}{2s^2}}, \tag{2}$$

where $\mathbf{x}_i$ is the i-th basis center (blue points in Figure 7) and $\mathbf{x}$ is a sample point in S_v (green points in Figure 7). In our experiments, we perform stratified sampling with each stratum containing 0.025% pixels of S_v and set s as $5\times$ the average sample distance.

Goal The general goal is to find proper $\mathbf{c} = \{\mathbf{c}_i\}$ and $\mathbf{T}$ so that the mapping $\mathbf{f}$ is as globally conformal and locally isometric as possible. In general, S_v is larger than S_r. To allow folding S_v into S_r, $\mathbf{f}$ should be surjective but not

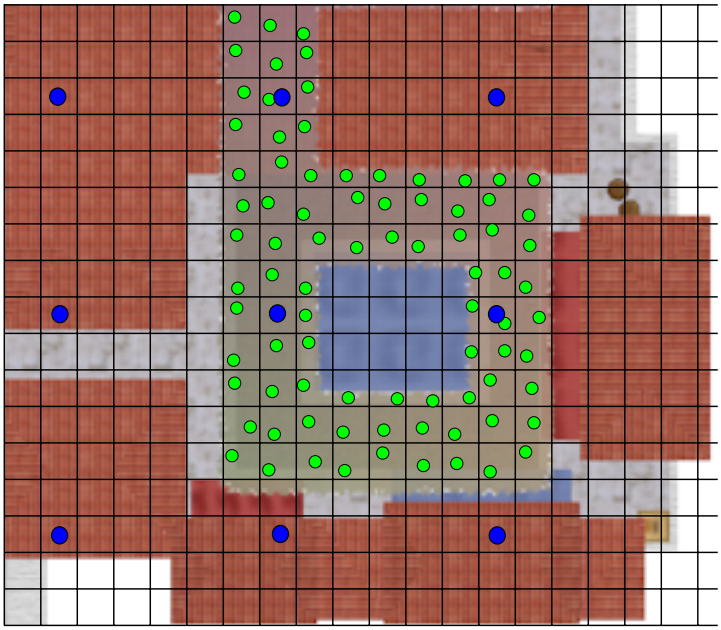


Figure 7: Stratified sampling example for part of the Italy scene floor plan.

necessarily bijective which is the goal of most of other planar mapping methods. Such folding will also prevent $\mathbf{f}$ from being globally-isometric. Thus, we target our ideal mapping as globally conformal but *locally* isometric, via a collection of objectives and constraints described below.

Conformal objective As 2D mappings satisfy the Cauchy-Riemann function when it preserves angles [24, 55], we define the conformal objective as:

$$\mathbf{E}_{conf}(\mathbf{c}) = \max_{\mathbf{x}} \left(\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 \right). \tag{3}$$

We then minimize this energy (i.e., a minimax formulation) to maintain smooth energy distribution without extra regularization.

Distance constraint Unlike a global isometric mapping which requires $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 1$, our mapping only needs to be locally isometric, which requires its Jaco-

bians $\mathbf{J}$ to satisfy $\mathbf{J}^T\mathbf{J} = \mathbf{1}$, that is,

$$\begin{aligned} \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 &= 1 \\ \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 &= 1 \\ \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y} &= 0. \end{aligned} \tag{4}$$

Since local isometry maps geodesics to geodesics [56], it suffices for VR locomotion. Note that minimizing $\mathbf{E}_{conf}$ in Equation (3) also addresses the last term in Equation (4), so we only need to focus on the first two terms.

Analogous to feature-aware texturing [57], different virtual regions may need different amounts of distance preservation in VR applications. For example, distances near region boundaries should be more strictly preserved as the users can examine the virtual walls close by, than when the users are in the middle of a large empty space. Due to this practical consideration, instead of putting the first two terms in Equation (4) as objective functions, we treat them as bounded constraints for more flexible control:

$$\begin{aligned} \alpha(\mathbf{x}) < \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 &< \beta(\mathbf{x}) \\ \alpha(\mathbf{x}) < \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 &< \beta(\mathbf{x}), \end{aligned} \tag{5}$$

where $\alpha \in [0, 1]$ and $\beta \in [1, +\infty)$ are stretching ranges for each virtual scene point $\mathbf{x}$. When both α and β equal to 1, the mapping is strictly locally isometric. However, for better conformality, we can relax the isometry into a range: the lower/higher the α/β value is, the more shrinking/stretching is allowed. There are three ways to set those parameters: constant values, user specification, or automatically computed via RANSAC line detection over S_v.

Exterior boundary constraint To keep all $\mathbf{u}$ inside the real space S_r, we construct the polygonal convex hull of S_r as a set of straight line functions $\{\mathbf{B}_i\}$, and add a series of linear constraints:

$$(\mathbf{B}_i\mathbf{u})^T(\mathbf{B}_i\mathbf{C}_r) > 0, \tag{6}$$

where $\mathbf{C}_r$ is the center of the physical space. The idea is to keep $\mathbf{u}$ and $\mathbf{C}_r$ on the same side of each $\mathbf{B}_i$ for testing point inclusion.

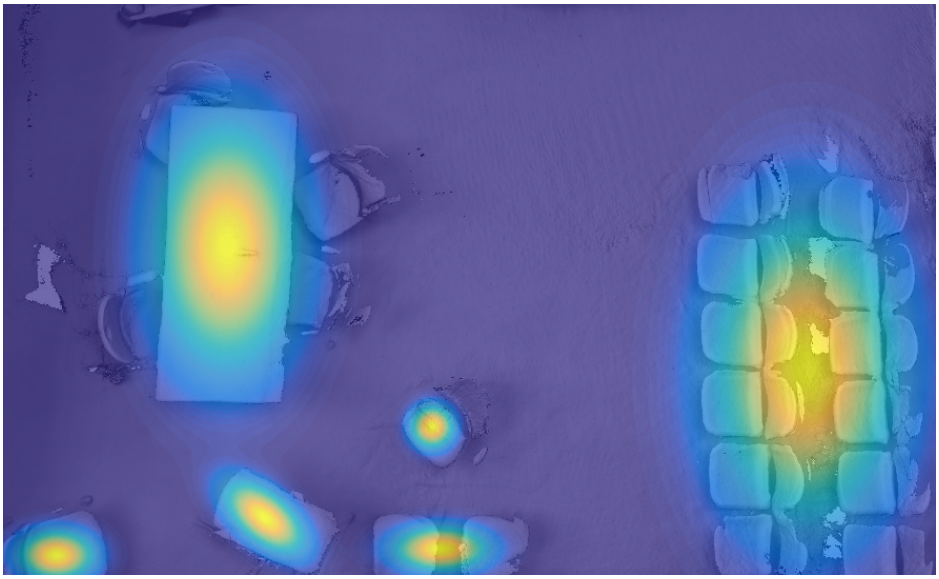


Figure 8: Energy distribution of the obstacle barrier in Equation (7) over a reconstructed real indoor scene from [2]. *Notice the higher energies in obstacle areas as indicated by brighter colors.*

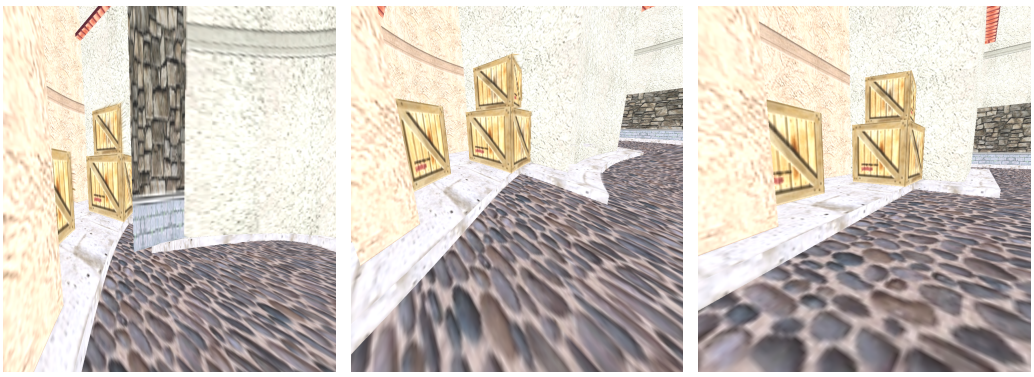
Interior obstacle barrier Preventing users from hitting interior obstacles can be formulated as the opposite of the point inclusion test in Equation (6). However, such formulation will require the calculation of directed areas or angles and solving a large quadratic instead of linear constraint system. For faster computation, we instead use a 2D Gaussian based barrier function for each interior object. For each object, we fit a minimal-covering 2D ellipse area $E(\mathbf{u}_c, E_w, E_h, \theta_c)$, where $\mathbf{u}_c$ is the center, E_w and E_h are width and height, θ_c is the rotation angle. Based on the scale of the ellipse, we define a Gaussian-based barrier:

$$E_b\left(E(\mathbf{u}_c, E_w, E_h, \theta_c), \mathbf{u}\right) = \exp\left(\frac{-1}{2\sigma^2}\left(\frac{u'^2}{E_w^2} + \frac{v'^2}{E_h^2}\right)\right), \tag{7}$$

where

$$\mathbf{u}' = \left(\mathbf{u}\begin{bmatrix}\cos\theta_c & \sin\theta_c \\ -\sin\theta_c & \cos\theta_c\end{bmatrix} - \mathbf{u}_c\right). \tag{8}$$

In our experiment, we set $\sigma^2 = 0.2$. Figure 8 depicts an example.



(a) *no local bijection* **(b)** *with local bijection* **(c)** *virtual view*

Figure 9: Local bijection. *Without local bijection, local fold-over in the static mapping may block the whole path with a wall (a). Adding the local bijection constraint can help prevent this artifact (b). (c) shows the original virtual scene for comparison.*

Local bijectivity Our mapping $\mathbf{f}$ allows global surjectivity to fold large S_v into small S_r. However, a local fold-over may produce visible artifacts, as exemplified in Figure 9. To prevent such fold-overs, we add a *local* bijectivity control, as described below.

A mapping [54] at a given point $\mathbf{x} \in S_v$ is locally bijective (i.e., no fold-overs) when it satisfies:

$$\det(\mathbf{J_u}(\mathbf{x})) > 0. \tag{9}$$

Directly applying this constraint to all points in S_v can be computationally expensive. More efficient barrier functions and optimizers [54] require sparse objective functions, whereas our objective function is dense. The method [22] can express Equation (9) as eigenvalues over all points, but such constraints cannot improve performance in our non-convex quadratic constraint problem.

To address this performance issue, we add local bijective constraints in a coarse-to-fine process. At the beginning, we partition S_v into a collection of cells (Figure 7). Then, during optimization, we add the following constraints to each sample point $\mathbf{x} \in S_v$:

$$\det\left(\mathbf{J_u}(\mathbf{x})\right) = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u} > 0. \tag{10}$$

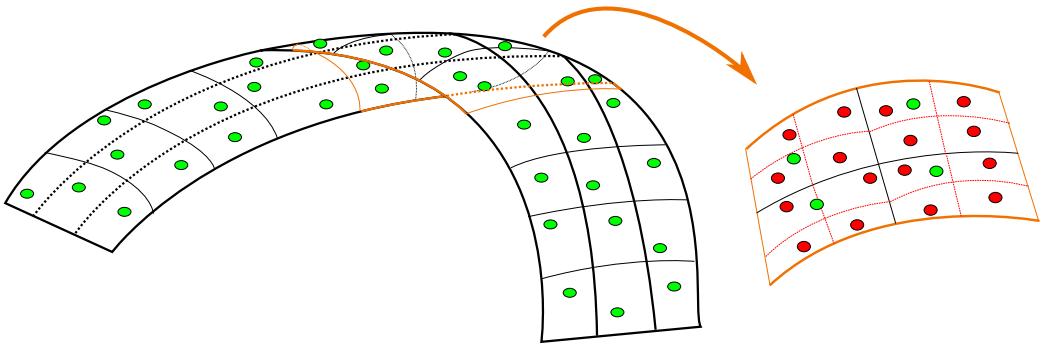


Figure 10: Local bijection sampling. *When fold-over is detected in the orange area, we restart the optimization, split all sample grids and add more samples. Distance constraints are then relaxed in the folding area.*

After convergence, if we find fold-over inside any cell, we iteratively split all cells into four smaller ones and add one more sample for each. We split all cells instead of only those in the fold-overs for faster convergence. Specifically, in our experiments we have found that up-sampling only the fold-over areas tends to push fold-overs to other areas with original sampling, which will require even more optimization rounds to fix than up-sampling all cells. Compared with active-set methods [22], the coarse-to-fine process is more stable for a non-convex problem.

Relaxed distance constraint To facilitate local bijectivity, we relax the distance constraints in Equation (5) to encourage stretching over folding. Here, imagine the virtual domain is a plastic floor plan sheet that can be bent or fold, but never cut. Intuitively, bending will cause point-wise stretching but folding will not. Thus, to encourage bending over folding, we maintain an extra point set L from those samples in a folding area, that is, the red points in Figure 10. We increase the upper limit of Equation (5) for all points in L to encourage stretching:

$$\begin{aligned}\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2<\beta(\mathbf{x})\rightarrow\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2&<\lambda\beta(\mathbf{x})\\ \left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2<\beta(\mathbf{x})\rightarrow\left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2&<\lambda\beta(\mathbf{x}).\end{aligned} \tag{11}$$

We set $\lambda = 1.2$ in our experiments.

Solver The local isometric requirement in Equation (5) makes the terms quadratic and thus cannot be directly solved via the SOCP methods (as in [22, 24]). With the combined conformal objective and various constraints and requirements, the problem becomes a quadratically constrained quadratic programming (QCQP) in a minmax format. However, due to the dual-bounded constraints Equations (4) and (5), the constraints are not convex and thus not suitable for QCQP solvers.

To address this large, dense, and non-linear optimization problem, we adopt an interior-point method [58]. In order to match the solver format, we rewrite the conformal objective Equation (3) as follows:

$$\min z, s.t. \left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 < z. \tag{12}$$

Combining all those constraints and barriers we discussed above, the final static mapping problem to be solved by interior-point method becomes:

$$\min z + w_o E_b(E, \mathbf{u}), \tag{13}$$

where w_o is the weight for obstacle barrier function, which we set as 600. We initialize $\{\mathbf{c}_i\}$ and $\mathbf{T}$ in Equation (1) as zeros and an identity matrix to satisfy Equations (5), (6) and (10) to (12).

3.2.2 Dynamic Inverse Mapping

The static map in Section 3.2.1 forwards positions from the virtual world S_v to the real world S_r. However, for VR walkthroughs we need the reverse map, from the current user position in S_r to S_v, as visualized in Figure 11. This reverse map needs to deal with the fact that the forward map might not be bijective and thus there can be multiple solutions. In addition, it should also minimizes perceptual angle and distance distortion during navigation. Below, we describe how we compute this inverse map dynamically during user navigation.

Start Given the user positions $\mathbf{u}(t)$ and $\mathbf{u}(t+1)$ as well as orientations $\mathbf{U}(t)$ and $\mathbf{U}(t+1)$ tracked in the real world S_r at time steps t and $t+1$, and the corresponding virtual position $\mathbf{x}(t)$ and orientation $\mathbf{X}(t)$ at time t, our goal is to compute the corresponding virtual position $\mathbf{x}(t+1)$ and orientation $\mathbf{X}(t+1)$.

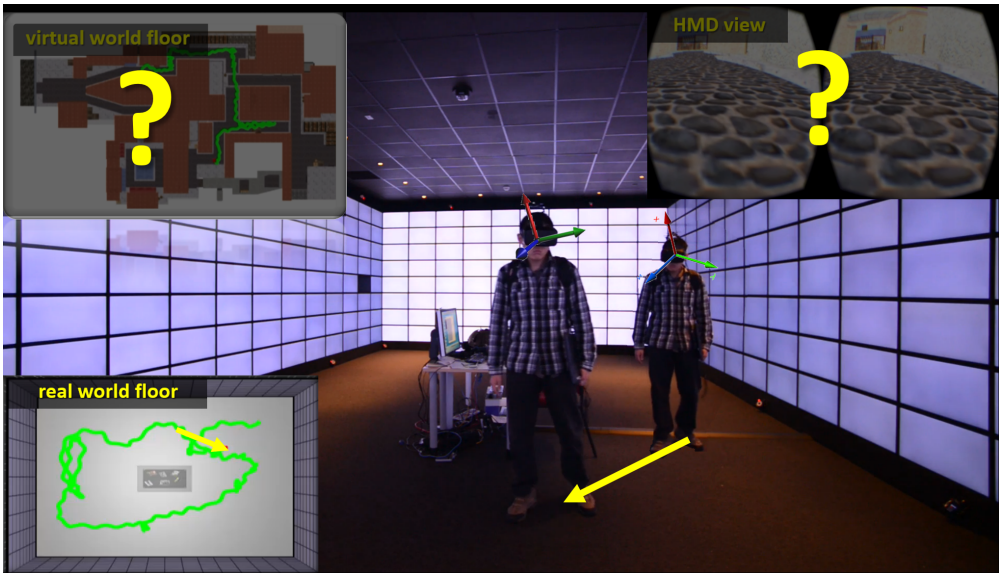


Figure 11: Dynamic reverse mapping illustration. *When users move and rotate the headset in real-time, we dynamically update the virtual camera position accordingly.*

Note that this is a path dependent process as $\mathbf{x}(t+1)$ and $\mathbf{X}(t+1)$ are computed from $\mathbf{x}(t)$, $\mathbf{X}(t)$, $\mathbf{u}(t+1)$, and $\mathbf{U}(t+1)$. We manually assign $\mathbf{x}(0)$ and $\mathbf{X}(0)$ for the initial virtual world position and orientation.

Direction update To compute $\mathbf{x}(t+1)$, we first compute the moving direction:

$$\hat{\delta}\mathbf{x}(t) = \frac{\mathbf{x}(t+1) - \mathbf{x}(t)}{\|\mathbf{x}(t+1) - \mathbf{x}(t)\|} \triangleq \begin{pmatrix} \hat{\delta}x \\ \hat{\delta}y \end{pmatrix}. \tag{14}$$

The virtual and real world directions are related by the Jacobians of their mapping:

$$\begin{pmatrix} \hat{\delta}x \\ \hat{\delta}y \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \begin{pmatrix} \hat{\delta}u \\ \hat{\delta}v \end{pmatrix}, \tag{15}$$

where

$$\begin{pmatrix} \hat{\delta}u \\ \hat{\delta}v \end{pmatrix} = \hat{\delta}\mathbf{u}(t) = \frac{\mathbf{u}(t+1) - \mathbf{u}(t)}{\|\mathbf{u}(t+1) - \mathbf{u}(t)\|} \tag{16}$$

is the real world direction. Thus, the goal is to find the Jacoabian of the reverse function of $\mathbf{f}$ in Equation (1):

$$\mathbf{J_u}(\mathbf{x}) = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{bmatrix}. \tag{17}$$

Even though $\mathbf{f}$ might not be globally bijective, the local bijectivity (Section 3.2.1) satisfies the inverse function theorem [59] and allows us to compute the inverse Jacobian via:

$$\mathbf{J_u}(\mathbf{x}) = \mathbf{J_x^{-1}}(\mathbf{u}), \tag{18}$$

where $\mathbf{J_x}(\mathbf{u})$ can be computed from the analytic function $\mathbf{f}$ at position $\mathbf{x}(t)$.

Position update We next compute the new virtual position $\mathbf{x}(t+1)$ based on the estimated direction $\hat{\delta}\mathbf{x}(t)$. We focus on the 2D x-y position, as the z/height value of $\mathbf{x}$ can be directly assigned from $\mathbf{u}$ after an initial correspondence. For computation purposes, we define $\mathbf{\Delta x}(t) = \mathbf{x}(t+1) - \mathbf{x}(t)$, and represent it in a polar coordinate system, that is, $\mathbf{\Delta x}(t) = \mathbf{\Delta x}_t(d, \theta) = (d\cos(\theta), d\sin(\theta))$. The goal is to find optimized (d, θ) to minimize an energy function as follows.

The first energy term measures how close the actual direction is to the estimated direction $\hat{\delta}\mathbf{x}(t)$:

$$E_{dir}(\theta) = \left\| \theta - \arctan\left(\frac{\hat{\delta}y}{\hat{\delta}x}\right) \right\|^2. \tag{19}$$

The second term is to keep the virtual distance close to the real distance:

$$E_{dis}(d) = \left\| d - \mathbf{\Delta u}(t) \right\|^2. \tag{20}$$

The last term is to match the mapping function $\mathbf{f}$ in Equation (1):

$$E_{map}(d, \theta) = \left\| \mathbf{f}\left(\mathbf{x}(t) + \mathbf{\Delta x}(t)\right) - \mathbf{u}(t+1) \right\|^2. \tag{21}$$

We find $\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{\Delta x}(t)$ to minimize

$$E_{rev} = E_{map} + \lambda_{dir} E_{dir} + \lambda_{dis} E_{dis}, \tag{22}$$

where λ_{dir} and λ_{dis} are relative weights. In our experiments, we set $\lambda_{dir} = 0.1$ and $\lambda_{dis} = 0.05$.

For fast convergence, we make the initial guess as:

$$\begin{aligned} \theta &= \arctan\left(\frac{\hat{\delta}y}{\hat{\delta}x}\right) \\ d &= \|\mathbf{\Delta u}(t)\| . \end{aligned} \tag{23}$$

Orientation update For rendering, we also need to compute virtual camera orientation $\mathbf{X}(t)$ from real camera orientation $\mathbf{U}(t)$, which is tracked by the HMD. We represent both orientations by their Euler angles:

$$\begin{aligned} \mathbf{U}(t) &= (yaw_{\mathbf{u}}(t), pitch_{\mathbf{u}}(t), roll_{\mathbf{u}}(t)) \\ \mathbf{X}(t) &= (yaw_{\mathbf{x}}(t), pitch_{\mathbf{x}}(t), roll_{\mathbf{x}}(t)) . \end{aligned} \tag{24}$$

Since our planar map $\mathbf{f}$ has only x-y positions, we compute only $yaw_{\mathbf{x}}$ and simply copy $pitch_{\mathbf{x}}$ and $roll_{\mathbf{x}}$ from $pitch_{\mathbf{u}}$ and $roll_{\mathbf{u}}$:

$$\begin{aligned} pitch_{\mathbf{x}}(t) &= pitch_{\mathbf{u}}(t) \\ roll_{\mathbf{x}}(t) &= roll_{\mathbf{u}}(t). \end{aligned} \tag{25}$$

A straightforward way to compute $yaw_{\mathbf{x}}(t)$ is copying the optimized angle θ from Equation (22). However, empirically we found that this may cause nausea and dizziness. To explain this, the static mapping (and thus the estimated orientation correspondence) is non-linear. Consequently, when users rotate their heads with uniform speed, the corresponding virtual camera may rotate non-uniformly. We thus compute $yaw_{\mathbf{x}}$ as a combination of λ_a and $yaw_{\mathbf{u}}$ to balance between accuracy and consistency:

$$\begin{aligned} yaw_{\mathbf{x}}(t) &= \lambda_a \theta(t) + \lambda_c yaw_{\mathbf{u}}(t) \\ 1 &= \lambda_a + \lambda_c \end{aligned} , \tag{26}$$

where λ_a and λ_c are subjective parameters set via user evaluation, as discussed in Section 3.3.1.

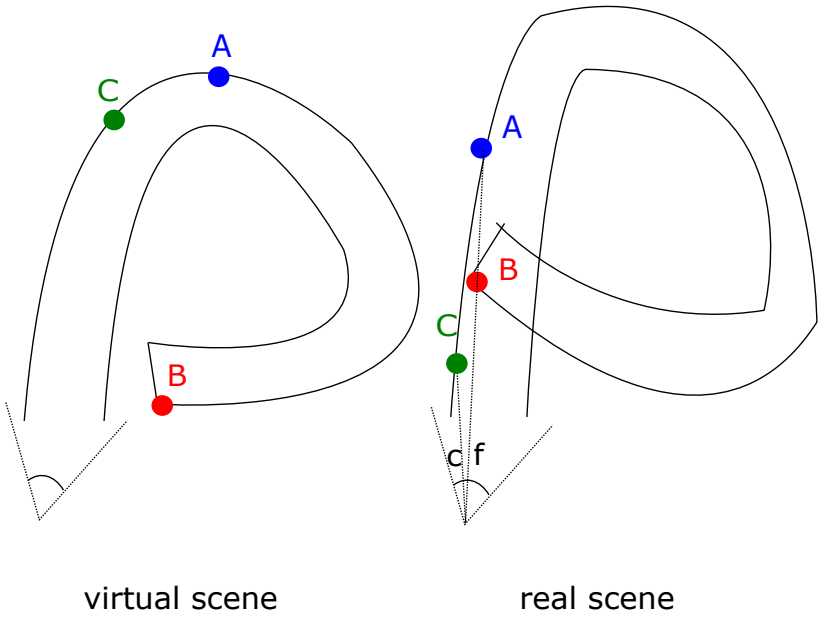
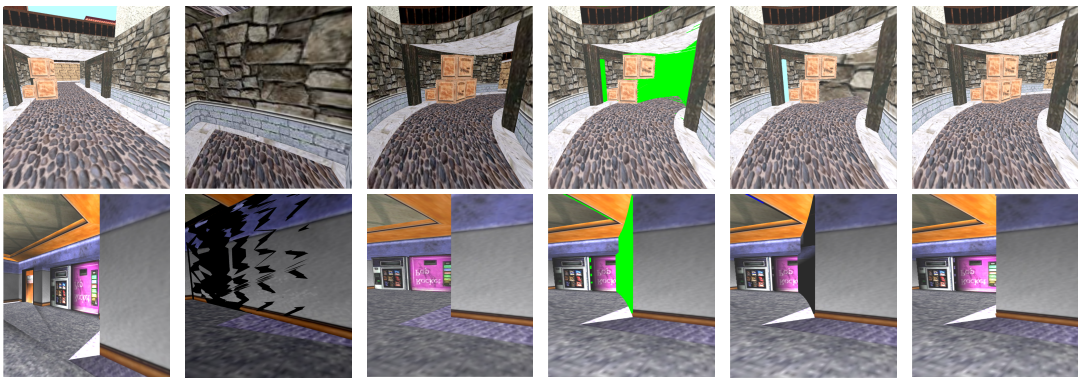


Figure 12: Handling occlusion in rendering. *The left and right are the virtual and real scenes. A camera is looking down a walled path, with three corresponding scene points shown in different colors. Point A is occluded in the virtual camera $\mathbb{C}_v$ but visible (i.e., dis-occluded) in the real camera $\mathbb{C}_r$. The goal is to decide how to render this dis-occluded pixel f in $\mathbb{C}_r$. Direct rendering the real scene G_r into $\mathbb{C}_r$ will pick B and cause highly occluded geometry in general (Figure 13b). The naive method improves this problem somehow (Figure 13e), but still picks point B for f because it is nearer the center of $\mathbb{C}_v$ than A. Thus, for each pixel in the dis-occluded area such as f, we search for its nearest non-occluded pixel in $\mathbb{C}_r$, which is c in this example. We then find the corresponding scene point C of pixel c, and assign its 3D distances to A and B in the virtual scene G_v as their depth values. Here, since C is closer to A than B, A will be picked for f. This strategy works because the local bijectivity of our mapping will prevent the incorrect solution B to be closer to C than the correct solution A.*

3.2.3 Rendering

From the tracked/computed real/virtual user positions/orientations (Section 3.2.2), we have the real and virtual cameras $\mathbb{C}_r = \{\mathbf{u}(t), \mathbf{U}(t)\}$ and $\mathbb{C}_v = \{\mathbf{x}(t), \mathbf{X}(t)\}$



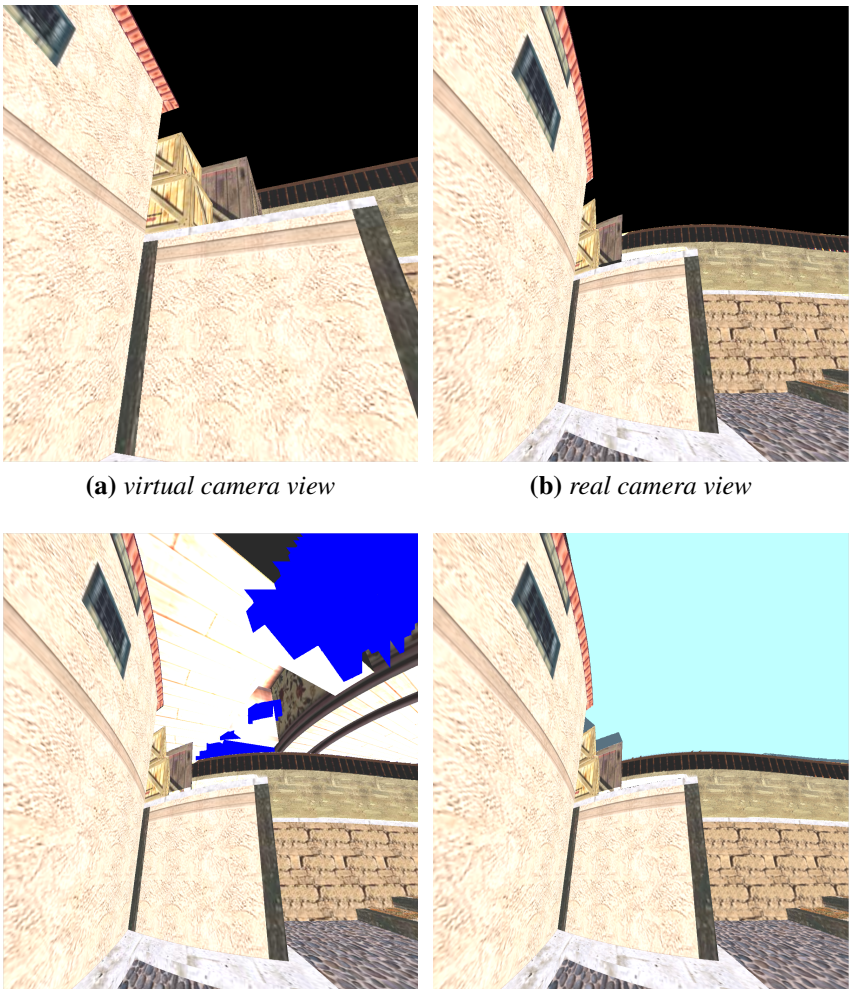
(a) *virtual view* **(b)** *real view* **(c)** *local view of (b)* **(d)** *warp + dis-occlusion* **(e)** *naive solution* **(f)** *our method*

Figure 13: Rendering examples. *(a): virtual camera* $\mathbb{C}_v$ *rendering* $\mathbf{I}_v$*. (b): real camera* $\mathbb{C}_r$ *rendering* $\mathbf{I}_r$*; notice the ghosting objects blocking most of the views. (c): manually cropping only the relevant parts of the real scene to remove the ghosting objects in (b); this is for comparison only, not for practical use. (d): warping* $\mathbf{I}_v$ *towards* $\mathbb{C}_r$ *with dis-occlusion areas visualized in green color. (e): naive solution for the dis-occlusion in (d) using fragment depth values in the virtual camera* $\mathbb{C}_v$*; however, this is also not enough. (f): our method fitting the virtual image* $\mathbf{I}_v$ *into the real camera* $\mathbb{C}_r$*; notice the combination of rendering in (a) and geometry in (c).*

at each moment t. Our goal is to render the appearance of the virtual world into the environment of the real world, so that users can perceive the former while navigating in the latter. As visualized in Figures 12 and 13, direct mapping the virtual scene geometry into the real scene via $\mathbf{f}$ in Equation (1) can have overlaps and intersections and thus not suitable for rendering. The original virtual scene rendering, however, cannot be used for direct navigation as it would cause motion sickness due to incompatibility with the real scene. We thus fit the rendering of the virtual world into the geometry of the real world, as discussed below.

Algorithm We first render the virtual image $\mathbf{I}_v$ with virtual scene geometry G_v and virtual camera $\mathbb{C}_v$. We then initialize the real image $\mathbf{I}_r$ by mapping/warping [25] $\mathbf{I}_v$ into $\mathbb{C}_r$ via $\mathbf{f}$ to maintain visibility consistency with $\mathbf{I}_v$. Parts of $\mathbf{I}_r$ might remain uncovered due to dis-occlusion, for which we perform another rendering pass via the real scene geometry G_r and camera $\mathbb{C}_r$.

Note that it is important to warp the original virtual image $\mathbf{I}_v$ into $\mathbf{I}_r$ first, followed by rendering G_r only into the uncovered portions of $\mathbf{I}_r$. Otherwise, parts of G_r may erroneously occlude points in $\mathbf{I}_v$ that should remain visible in $\mathbf{I}_r$, as demonstrated in Figure 13b. For the dis-occluded parts in $\mathbf{I}_r$, simply rendering G_r via $\mathbb{C}_r$ will show artifacts in Figure 13b in these dis-occluded regions. A naive method that partially improves the outcome is to use depth values in the virtual camera $\mathbb{C}_v$ instead of the real camera $\mathbb{C}_r$. However, this is still not enough, as demonstrated in Figure 13e. Thus, we propose a more accurate method for dis-occlusion areas as follows. For each fragment (xy pixel + z depth) f_r visible in $\mathbf{I}_r$ but not in $\mathbf{I}_v$ (i.e., in dis-occlusion areas), we find the nearest (in terms of 2D xy coordinates) fragment f_v visible (i.e., non-occluded) in $\mathbf{I}_r$. We then assign the 3D Euclidean distance between f_v and f_r (in $\mathbb{C}_v$ space) as the depth value to f_r for rendering in $\mathbb{C}_r$. Figure 12 illustrates an example. Intuitively, this strategy works because the local bijection in Section 3.2.1 prevents overlapping among nearby virtual scene points. Notice the geometric similarity between the local real scene in Figure 13c and our result in Figure 13f.



(a) *virtual camera view* **(b)** *real camera view*

(c) *without environment map* **(d)** *with environment map*

Figure 14: Importance of the environment map, a sky blue background in this example. *(a): rendered from the virtual camera view. (b): rendered from the real camera view. (c): our method without using environment map; notice the ugly artifacts in the sky. (d): our method with environment map; the sky becomes clear, similar to (b).*

Implementation Our rendering algorithm is amenable for GPU implementation. We first render the virtual image $\mathbf{I}_v$ (via polygon rasterization) of the virtual geometry G_v into virtual camera $\mathbb{C}_v$. For each pixel/fragment, we record the usual

color and depth, as well as the optional G-buffer parameters [60] for re-shading non-Lambertian materials. We then forward warp [25, 26] $\mathbf{I}_v$ into $\mathbb{C}_r$ to initialize $\mathbf{I}_r$ (and optionally re-render non-Lambertian fragments), and record the coverage via a stencil buffer. If $\mathbf{I}_r$ is not entirely covered, we render G_r via $\mathbb{C}_r$ culled by the stencil buffer. Because it is computationally complex to do reverse projection of $\mathbf{I}_v$ as $\mathbf{f}$ is a transcendental function, we store the 3D position of a real rendered pixel as a texture coordinate.

Similar to standard game level design, we surround the scene with an environment-map box to ensure all pixels in $\mathbf{I}_v$ are initially covered. Thus, all uncovered pixels in forward-warped $\mathbf{I}_r$ are caused by dis-occlusion. The environment map is important to ensure robust dis-occlusion to prevent far-away objects being mistakenly rendered into the background, as exemplified in Figure 14.

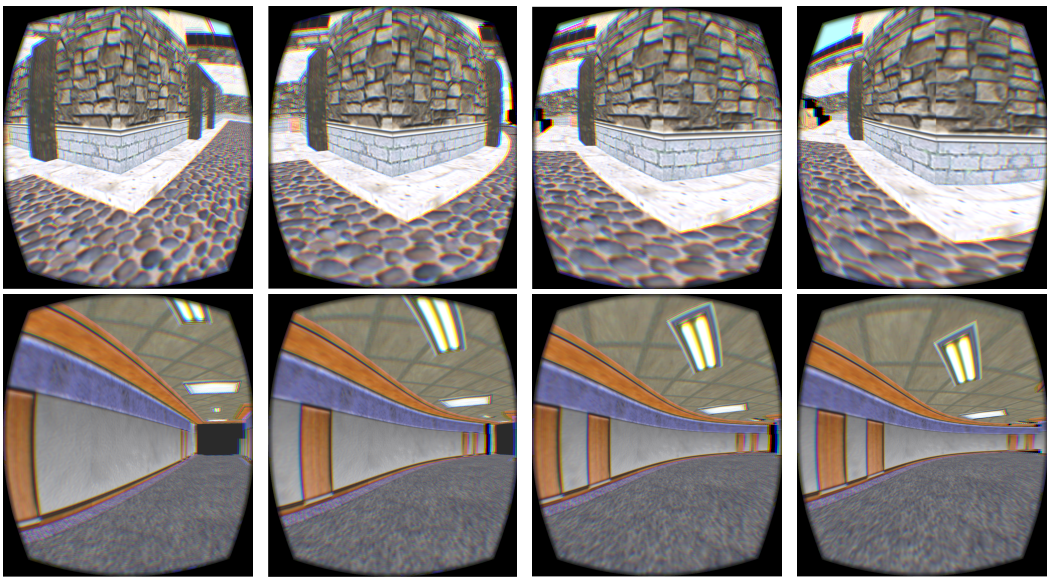
For more robust handling of larger/more-complex dis-occlusions, for each dis-occluded fragment f_r we find its four instead of just one nearest visible fragment, as described above. Specifically, we find the nearest non-occluded pixel $c_i, i \in [1, 4]$ visible (i.e., non-occluded) along each image domain direction ($\pm x$ and $\pm y$) in $\mathbf{I}_r$. Each of them has 2D distance d_i to f_r and corresponding scene position C_i. We then estimate the hypothetical matching point C (as in Figure 12) by:

$$C = \frac{\sum_{i=1}^{4} d_i^{-1} C_i}{\sum_{j=1}^{4} d_j^{-1}}. \tag{27}$$

Discussion Conceptually, the mapping $\mathbf{f}$ in Equation (1) can have a meta parameter w that morphs the scene [61]: $\mathbf{f}_0$ maps to the virtual scene $\mathbf{G}(0) = G_v$, $\mathbf{f}_1$ maps to the real scene $\mathbf{G}(1) = G_r$, and $\mathbf{f}_w$ with $w \in (0\ 1)$ maps to a scene in-between $\mathbf{G}(w)$. The w parameter trades off between visual fidelity to the virtual image $\mathbf{I}(0) = \mathbf{I}_v$ and motion fidelity to the real scene $\mathbf{G}(1) = G_r$. Figure 15 compares renderings with different mixing weights.

To improve quality, we follow the standard IBR tricks [62] of rendering multiple images with different camera parameters for G_r, and blend all fragments with the same screen position and depth (within numerical precision range) with weights proportional to their quality. For example, fragments with normal directions closer to the camera ray will have higher weights.

The camera $\mathbb{C}_r$ might go beyond the field of view of the original camera $\mathbb{C}_v$. We thus render $\mathbf{I}_v$ into a cylinder or 6 sides of a cube to ensure sufficient coverage.



(a) $w = 0\%$ **(b)** $w = 33\%$ **(c)** $w = 66\%$ **(d)** $w = 100\%$

Figure 15: Comparison of different virtual-real mixing weights w in rendering. *(Scene courtesy of Counter Strike, Italy (top) and office (bottom).)*

For efficiency, we estimate and render only the needed subset of $\mathbf{I}_v$ from $\mathbb{C}_v$ and $\mathbb{C}_r$, which includes the relevant cube faces stenciled with the necessary portions.

3.3 Evaluation

We have evaluated subjective and objective aspects of our system pipeline via various experiments. We have recruited 7 participants with ages between 25 and 31. Among these participants, one participant has no prior experience with HMDs, six others have at least some basic knowledge or experiences with HMDs. One of the participants suffered from a light degree of vertigo.

3.3.1 Subjective Parameters

Design In addition to objective parameters which are empirically set based on scene properties, we also have to evaluate two subjective parameters, w in Sec-

tion 3.2.3 and $\frac{\lambda_a}{\lambda_c}$ in Section 3.2.2. Due to limited human sensitivity to small parameter differences, we evaluate ranges instead of individual values. For w, we uniformly split its valid range $[0, 1]$ into 3 subranges. For $\frac{\lambda_a}{\lambda_c}$, values lower than 0.6 may cause significant mismatch between virtual and real camera orientations, triggering large dis-occlusions and rendering artifacts (Figure 19). We thus chose the $[0.6, 1]$ range and uniformly split it into 4 sub-ranges. For all experiments, each participant was tested with a random value sampled from each sub-range.

For w, we asked the participants to follow paths inside the virtual office scene (Figure 15) which are mapped to the physical lab scene with both boundaries and obstacles. Intuitively, larger w values will favor locomotion over visual fidelity, and w that is too small can cause motion sickness. We asked the participants to choose the most favorite w value to balance between visual and locomotion fidelity. Because people are more sensitive to locomotion discomfort, the participants were evaluated with high to low w values until feeling uncomfortable. Since $\frac{\lambda_a}{\lambda_c}$ is for orientation, we asked the participants to remain stationary, rotate their heads, and choose which values provide the most natural experience.

Result For w, 6 participants chose values within the 33% to 66% range as their preferences. One of those 6 users reported unbearable locomotion experience when w lies in 0 to 33%. The user with 3D vertigo reported light but bearable dizziness during this range and prefers the 66% to 100% range instead.

For $\frac{\lambda_a}{\lambda_c}$, all participants reported visual discomfort when the value is lower than 0.7. Among all participants, 1 chose $[0.7, 0.8]$ while the others chose $[0.8, 0.9]$. They reported that the proper values should be position-sensitive, that is, the level of mismatch between real and virtual scenes caused by $\mathbf{f}$.

We observed that participants did not rotate their head much during the w experiments. As a result, they did not detect strong inconsistency when λ_c is low until they were asked to do the rotation experiment.

3.3.2 VR Usability

Design Similar to other VR applications [63], we have conducted a formative user study to evaluate the usability of our VR system. We chose the task-based method to evaluate locomotion and the post-interview-based method to evaluate

visual fidelity and motion sickness. All subjective parameters in this study session are taken from users' choices described in Section 3.3.1.

For this experiment, we chose the Counter Strike office scene as it has simple and uniform paths for task-based studies. We picked the purple vending machine at one end of the scene as the goal object. This particular object is unique in the scene and far from the entrance door, and thus suitable for a search task. Without prior knowledge of the scene, the participants started free locomotion to find the target. During the experiment, if a participant hits any real world objects or feels sick, we stopped the experiment. To evaluate locomotion fidelity, the participants can click a counter whenever they feel lost. To evaluate visual fidelity, the participants were given the original virtual scene to explore and grade for similarity at the end of the experiments. At the closing interview session, the participants shared their comments with us and graded their sickness and fatigue levels.

Result For the search task, 2 participants had a failure experience. One was caused by wall crossing; since our current pipeline does not implement collision detection, when users cross a wall our algorithm will get stuck by regarding all pixels as dis-occlusion. Another failure case was caused by the particular user passing-by the vending machine without noticing it until being explicitly prompted. No participant hit any real world objects.

- On locomotion fidelity, no direction loss was reported.
- On motion sickness, we have conducted the simulator sickness questionnaire (SSQ) [64] at the end of the experiments. Compared to prior literature, such as [65] for gaming environments, our overall numbers as shown in Table 1 are in a comfortable range. Specifically, one user reported bearable fatigue after the w experiment. Two users (one with VR experience and another without) reported dizziness right after the first experiment, but they recovered and felt comfortable for the remaining two experiments.

Table 1: SSQ results using the questionnaire from Bouchard et al. [4].

participants / measure	1	2	3	4	5	6	7
nausea (out of 27)	5	8	6	1	0	0	0
oculomotor (out of 21)	5	6	4	1	1	1	0

- On visual fidelity, One participant graded 90 (out of 100), five graded 80, and one graded 40.

The reason for this latter low grade is that the participant is more sensitive to corner angles. Since our algorithm warps the virtual scene, global angle (not local conformality) might change significantly.

For additional interview comments, the participant without VR experience was excited about the walkthrough experience. Another participant expressed concerns about highly bended angles, which may cause users fatigue and discomfort after sustained usage.

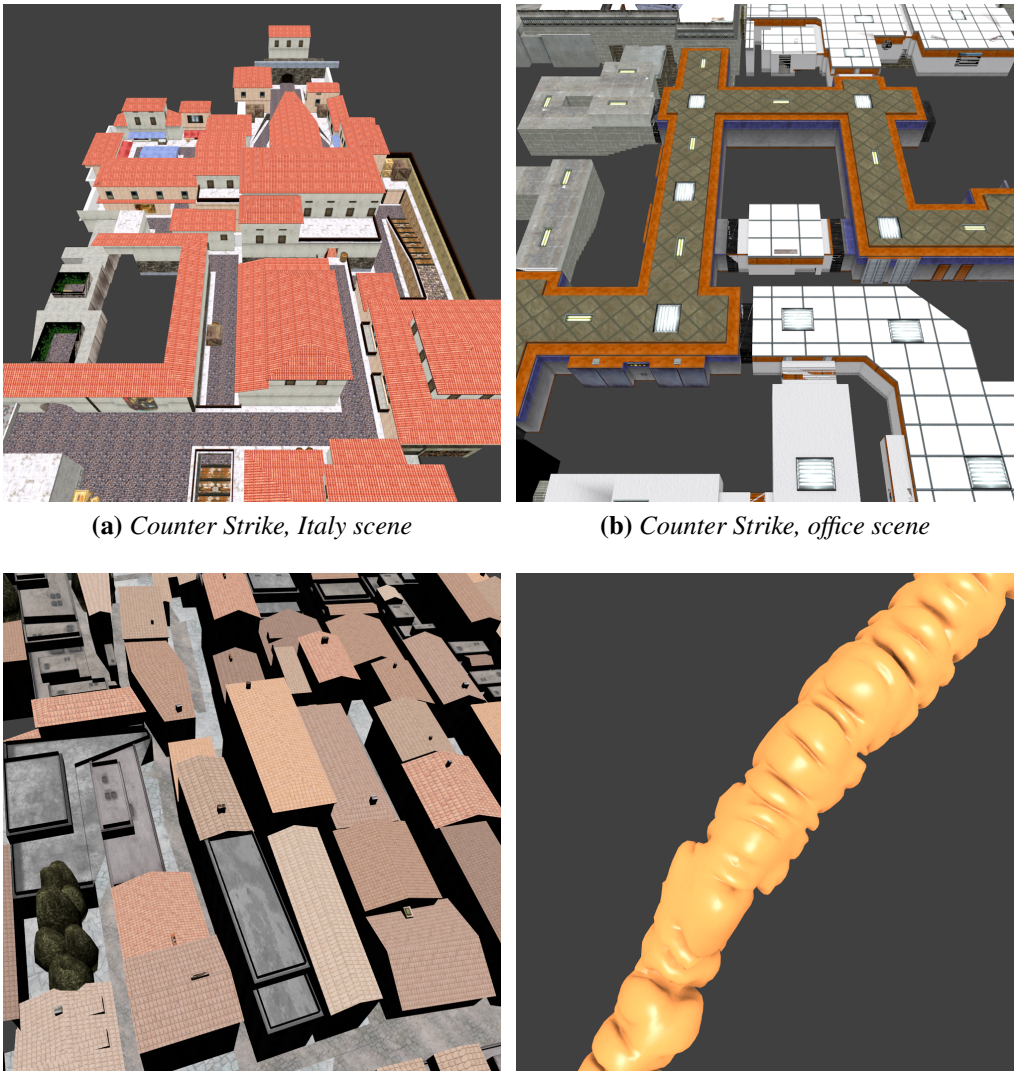
3.4 Applications

Our method can be applied for various VR applications. These include gaming/entertainment Figures 16a and 16b, architecture/design walkthrough Figure 16c, and medical imaging/visualization Figure 16d.

First person shooting games, such as Counter Strike, have been a main target for VR designers. We have applied our method to two different scenes in Counter Strike: the Italy scene for outdoor views and the office scene for indoor views, as shown in Figures 16a and 16b. The corresponding results in Figures 5 and 15 and the accompanying video show the promise of our method for navigating first person shooter games while moving in a real room using HMD.

Virtual architectural walkthroughs provide immersive experience for design and education. Figure 16c displays a portion of the Venice city 3D model, with our rendering result shown in Figure 17a. Using our system, users can virtually walk to every corner of the city while staying in their homes.

Modern virtual reality techniques have been introduced in medical imaging with clinical use. As a representative application, virtual colonoscopy is a non-invasive computer-aided 3D medical imaging technology to replace traditional optical detection [12]. To make radiologists feel being inside a human colon while having realistic navigation experience, we straighten a folded colon for 2D navigation Figure 16d. A sample rendering by our system can be seen in Figure 17b.



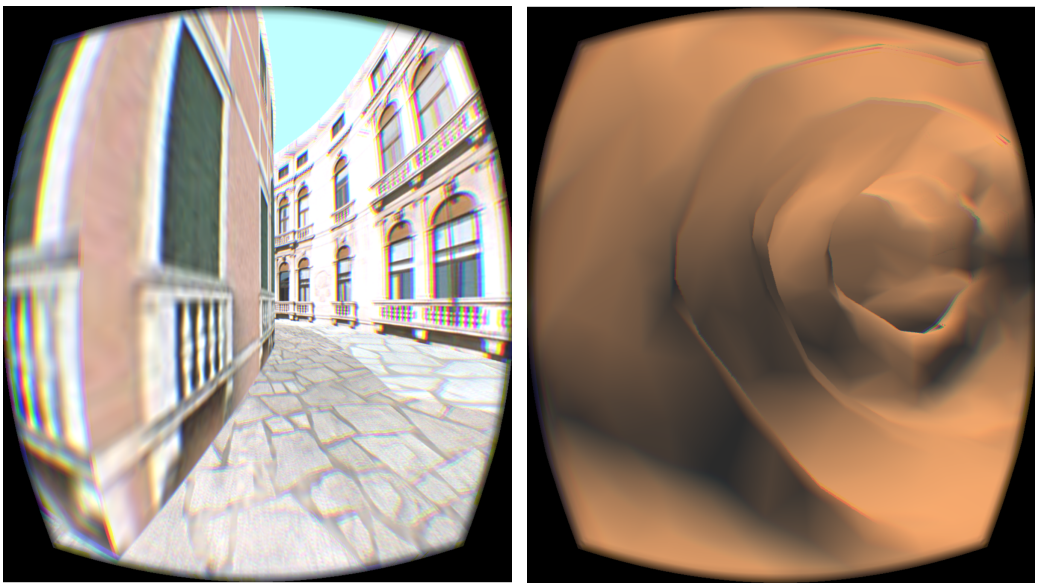
(a) *Counter Strike, Italy scene*

(b) *Counter Strike, office scene*

(c) *Venice scene*

(d) *virtual colonoscopy*

Figure 16: External views of all scenes used. *(a) and (b) are from Counter Strike game level data, (c) is freely available from tf3dm.com, and (d) is from our in-house anonymous patient database.*



(a) *Venice scene* **(b)** *virtual colonoscopy*

Figure 17: Sample HMD rendering for the Venice scene and virtual colonoscopy. *The corresponding results for the Italy and office scenes can be found in Figures 5 and 15.*

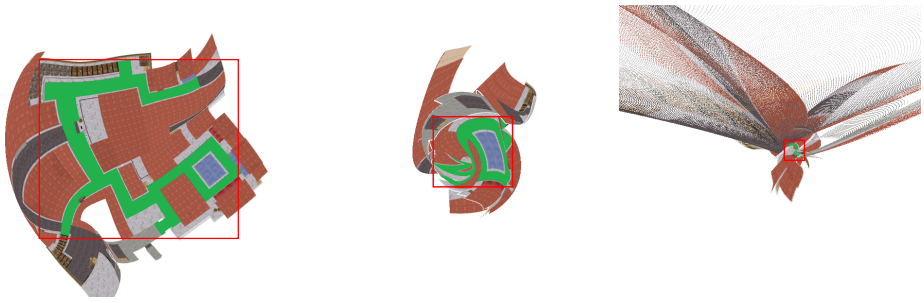
3.5 Conclusion

We have presented a method to support real walking within a given physical environment, while perceiving a given virtual world inside an HMD for VR applications. Our system focuses on the graphics aspects: 2D mapping and 3D rendering to balance between visual fidelity to the virtual world and locomotion comfort for the physical world. These graphics algorithms depend on but are largely orthogonal and complementary to a collection of perceptual and physiological parameters, for which we have performed a preliminary study but are definitely worth more thorough investigation. Some specific evaluations and extensions including a larger scale user study as well as the incorporation of physics laws and tactile interaction [18] are planned.

3.6 Limitations

Our current method is designed for mapping 2D floor plans. It can be used to navigate scenes with height variations by tracking user head elevation and add it over the virtual scene height value for rendering. This can work if the users actively change their body postures such as jumping or crouching, but they might not sense the proper motion with plain walking up a ramp.

Although our static mapping solver can estimate a constraint-satisfying solution for any given pair of virtual and real spaces, the quality depends on their shape and size differences. When their sizes differ substantially, the mapped virtual space may be strongly distorted, reducing both visual and locomotion fidelity to the users. On the other hand, similar virtual and physical worlds can certainly improve the output quality of our system. Please refer to Figure 18 for comparisons.



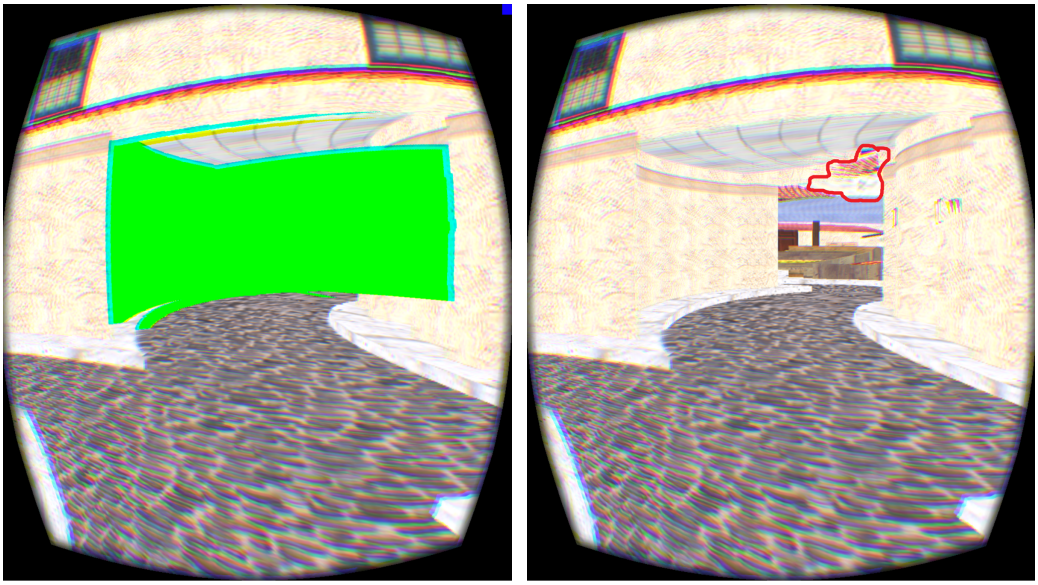
(a) *small* **(b)** *medium* **(c)** *large*

Figure 18: Differences in size/shape between the virtual and real spaces. *Our method can compute static mapping between a given pair of virtual and real scenes, but the quality depends on their size/shape differences. The red rectangles indicate the extents of the real spaces.*

Large open virtual spaces cannot be folded into a small real space without being noticed by the users. Space manipulation mechanisms such as teleportation might help and are worth further study. Fortunately, many virtual scenes in popular VR applications such as gaming, architecture, and medical imaging are highly occluded, and thus can benefit from our method.

In some cases, narrow virtual pathways can become even narrower than regular human footsteps after the static mapping. This, combined with motion capture accuracy limit, can cause our estimated $\mathbf{\Delta x}$ to cross walls and obstacles.

Our rendering algorithm cannot handle a large area that is entirely dis-occluded and also has very different depth from the surrounding regions, as shown in Figure 19. In particular, our dis-occlusion handling method, as depicted in Figure 12, requires parts of the nearby geometry to be visible for a dis-occluded fragment. Our experiments and analysis indicated that these dis-occlusion artifacts can be resolved by using more virtual images. Exactly how many additional virtual cameras are necessary, and where to place them, remain future research problems.



(a) *dis-occlusion area* **(b)** *artifacts*

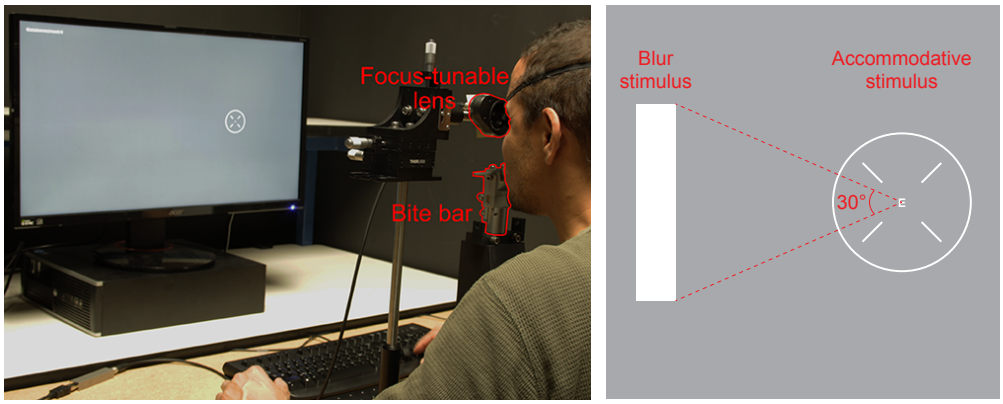
Figure 19: Limitation in our rendering algorithm. *(a) shows an entire dis-occluded area in green color, which has vastly different depth from the surrounding rendered areas as shown in (b), with the rendering artifact circled in red.*

During the initial phase of the project, we have explored the use of non-pinhole cameras [27,66] for rendering, but settled for traditional pinhole projection due to perceptual and performance reasons. In particular, pinhole projection is faster to render, and causes less motion disorientation for VR walkthroughs. However, a limited amount of non-linear projection is still worth further investigation to better balance between rendering and perception.

4 Accommodation

The human visual system can resolve higher spatial frequencies in the fovea than in the periphery. This property has been harnessed by recent 2D foveated rendering methods to reduce computation cost while maintaining perceptual quality. Inspired by this, we have conducted psychophysical experiments to study foveation of human visual systems for 4D light fields and evaluate our prototype system [10].

We measure, for the first time, the blur detection/discrimination and light field depth perception thresholds in up to 15 degree of visual eccentricity, and reject the idea of replacing the peripheral rendering with 2D billboards – 4D light fields are still required. The psychophysical data can also guide other foveated rendering approaches.



(a) *hardware setup* **(b)** *stimulus*

Figure 20: Experiment design to measure sensitivity to change in blur. *One eye of the participant is fixed on the center of the accommodation stimulus while the blur stimulus changes. We measure the thresholds of detecting or discriminating the blur changes under varying eccentricities and baselines.*

4.1 Perceptual Study of Visual Sensitivity

Our goal of rendering foveated light field is to sample the 4D information as succinctly as possible: using the fewest number of rays to represent objects at different depths and eccentricities without letting the user notice the differences. Addi-

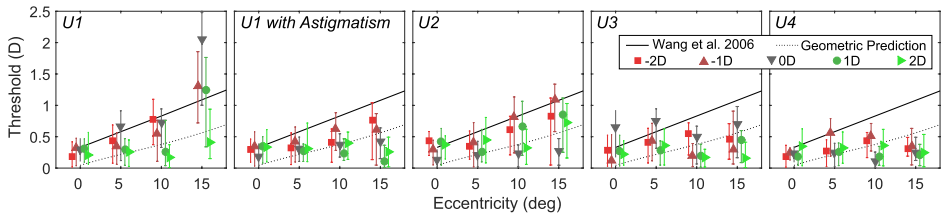


Figure 21: Blur perception study results. *These figures plot the thresholds of blur detection and discrimination measured as a function of eccentricity and pedestal/baseline blur* $(-2, -1, 0, 1, 2D)$ *for four different subjects. X-axis represents retinal eccentricity in degree. Y-axis represents thresholds measured at varying eccentricities and pedestal blurs. Each vertical bar indicates the* 75% *performance level centered at a* 95% *confidence interval. Threshold values increase with eccentricity for subjects U1 (both with and without correction for astigmatism) and U2 but not for U3 and U4. The measured thresholds were lower than both [67] and our geometrical prediction. This suggests a conservative saving within* $15\,\mathrm{deg}$ *of eccentricity.*

tional reduction in angular bandwidth may be possible if the visual system is not sensitive to focal cue in the periphery. An extreme case is to replace the peripheral 4D light field rendering with a 2D billboard, but the sensitivity and detection thresholds need to be found. The ultimate judge for this approach is our visual system. Specifically, we ask: are there conditions in which we can omit angular sampling without the visual system detecting it? We investigated this question by measuring the visual sensitivity to changes in focus cue. In Section 4.1.1, we measured blur discrimination thresholds where a focus-tunable lens changed the targets' focal power to generate blur. We found that the thresholds varied significantly across individuals; some reached very low values. In Section 4.1.2, we measured depth discrimination thresholds where the eye automatically generated blur using our display prototype. We observed that under light field display scenarios, our ability to discriminate depth from focal cue degrades more consistently as eccentricity grows. The difference motived us to further analyze and formulate the whole display-eye system. Please refer to our supplementary video for live captures.

4.1.1 Blur Perception with Optical Stimuli

Setup We have designed a psychophysical experiment to measure the blur detection and discrimination thresholds at different eccentricities: $0\,\mathrm{deg}$ for fovea, and 5, 10, and $15\,\mathrm{deg}$ for different peripheries. The setup is photographed in Figure 20. Blur pedestals, the baseline blur in the discrimination task, of $-2, -1, 0, 1,$ and $2D$ were tested at all eccentricities. The visual stimulus was presented on an LCD display (Acer XB270HU, 2560×1440 resolution, $144Hz$ refresh rate), at the desired location to control visual eccentricity. The display was located at 80 cm away from the viewer, where the central pixel subtended 1 arcmin from the viewer's eyes. A focus-tunable lens (Optotune, EL-16-40-TC, response time $30ms$), placed before one eye of the subject, controlled the focal distance of the stimulus. The field of view provided by the lens subtended $13\,\mathrm{deg}$ in radius. A bite bar was used to precisely position the viewer's eyes at the desired location.

Calibration Every subject went through a calibration procedure before starting the measurements. This calibration is necessary to make sure the viewer's eyes stayed at the farthest point of his/her accommodation. First, we found the farthest point of accommodation by a tumbling E test [68] and a staircase procedure [69]. Second, magnification and translation due to the change in focal power of the lens was quantified using alignment tasks.

Stimuli The visual stimulus for blur detection/discrimination was a bright rectangle ($100cd/m^2$) drawn on a dark background ($20cd/m^2$). The size of the foveal rectangle was 0.16 (W) $\times$ 0.8 (H) deg. The size of the peripheral rectangles scaled linearly with visual eccentricity - 0.04 (W) and 0.2 (H) $\times$ the eccentricity deg. The focus-tunable lens operated with the display to introduce defocus blur to the rectangles. The presentation time for the rectangle was kept to $0.3\,\mathrm{sec}$, short enough to prevent accommodation.

The rectangle appeared twice in a random sequential order with different amounts of blur; one with pedestal blur for discrimination (or no blur for detection) and the other with more blur in addition to the pedestal blur. Fixation target was inserted for $0.5\,\mathrm{sec}$ between the two intervals to discourage subjects from accommodating to the stimulus. The task was a 2-alternative-forced-choice: subjects chose the one that appeared blurrier and had to guess when not sure.

More than 100 trials were executed per combination of blur pedestal and visual eccentricity. Total duration of the experiments including calibration and training was about 6 hours.

Subjects Four subjects, aged 31 to 48, participated. All subjects had normal or corrected-to-normal visual acuity. One subject was an author. The other three subjects were unaware of the experimental hypothesis.

Measurements The measurement results are shown in Figure 21. The Y-axis represents blur discrimination thresholds. Each bar indicates the 75% performance level centered at a 95% confidence interval. We estimated threshold values by fitting a cumulative Gaussian function to the performance curve drawn as a function of size of differential blur [70]. Specifically, we measured performance at many values of added blurs. We then estimate participants' performance levels as a function of the magnitude of added blur. The performance level will be 50% (chance level) when user cannot tell the difference, and it will be 100% when user can tell. Note that performance level is different from whether user can see the difference. Performance of 50% corresponds to 0% of seeing the difference. Similarly, performance of 75% corresponds to 50% of seeing the difference. We used a statistical method, psychometric function fitting [70], to estimate where the 75% performance point. Threshold is defined to be the magnitude of blur where performance of 75% occurs.

Results The experimental results are shown in Figure 21, where baseline defocus is color coded. Note that $0D$ baseline means blur detection threshold. The dashed black line shows geometrically estimated detection thresholds by comparing a cylindrical blur kernel for a $5mm$ pupil and ganglion cell densities. Solid black line shows the prediction curve suggested by Wang et al. [67]. Both lines are plotted for theoretical comparison with our collected data. The results show three observations. First, the thresholds increased as a function of eccentricity for some subjects but not all; the threshold values for two subjects (U3 and U4) remained nearly constant and below the theoretical curves at farther eccentricities. Second, correction for astigmatism of one subject (U1, astigmatism$= -1.25D$ at $8\,\mathrm{deg}$) did not significantly improve sensitivity to blur change, if not harmed. Third, varying baseline yielded large thresholds differences given an eccentricity. This may be attributed to various factors, e.g., peripheral refractive state [71].

4.1.2 Depth Perception with Light Field Display

Many studies showed our focal cue sensitivity decreases with retinal eccentricity. This includes blur size change, depth perception, and accommodative responses. However, light field displays provide focal cues differently from nature because the eye receives only a finite number of discrete rays. To our best knowledge, no prior work studied the eccentricity effect on light field depth perception.

We measured monocular depth perception, a key factor differentiating between light field and conventional 3D displays. With our prototype (Section 5.5), we first conducted a pilot study to verify the existence of foveation from light fields. Motivated by the results, we then performed a thorough psychophysical experiment for depth detection thresholds along eccentricities.

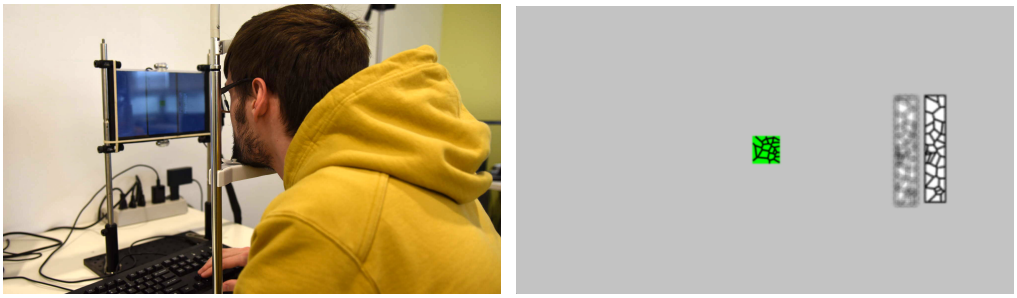
Design Figures 22a and 22b show the study design. Subjects were seated at $3.3D$ ($30cm$) from the light field display with their non-dominant eye occluded. A chin rest was used to accurately control the viewing distance.

The stimulus consisted of one fixation target and two test targets, both have broadband binary Voronoi diagrams textures (Figure 22b). The fixation target at $3.8D$ was a small green square. It remained at the center of the screen to fix subjects' gaze and focal distance. Subjects were asked to keep watching this fixation target during the entire study to keep both gaze position (i.e., fovea center) and focal distance invariant. The test targets were two vertically elongated rectangles rendered side-by-side with a small gap $2.5mm$ (to avoid occlusion cue). The first test target was at same depth of the fixation but rendered behind to avoid occlusion. The second object was rendered at $2.8D$. They appeared at one of 8 eccentricities from the fovea to $15\,\mathrm{deg}$.

We ran 20 trials for each of these 2 conditions. In each trial, users were asked to select which test target looked closer. We used the method of adjustment to measure the depth detection thresholds. At the beginning of each trial, the two test targets were positioned at the same depth ($3.2D$). Then subjects pressed up/down arrow buttons to increase/decrease the two targets' depth separations until reaching the thresholds where they can perceive the relative depths. We dynamically rescaled the sizes of test targets based on their depths so that they always appear the same size (to avoid size cue). Subjects were warned when the relative depth reached 0 or the hardware limit. Each eccentricity contains 4 trials. All trials were

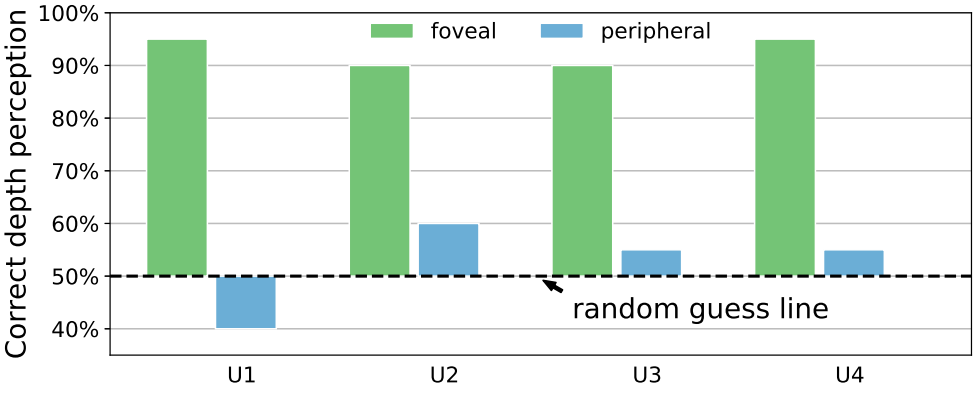
randomized.

Results Figure 23b plots the mean thresholds with standard deviation. All subjects showed consistent trend of increasing thresholds with respect to eccentricity.

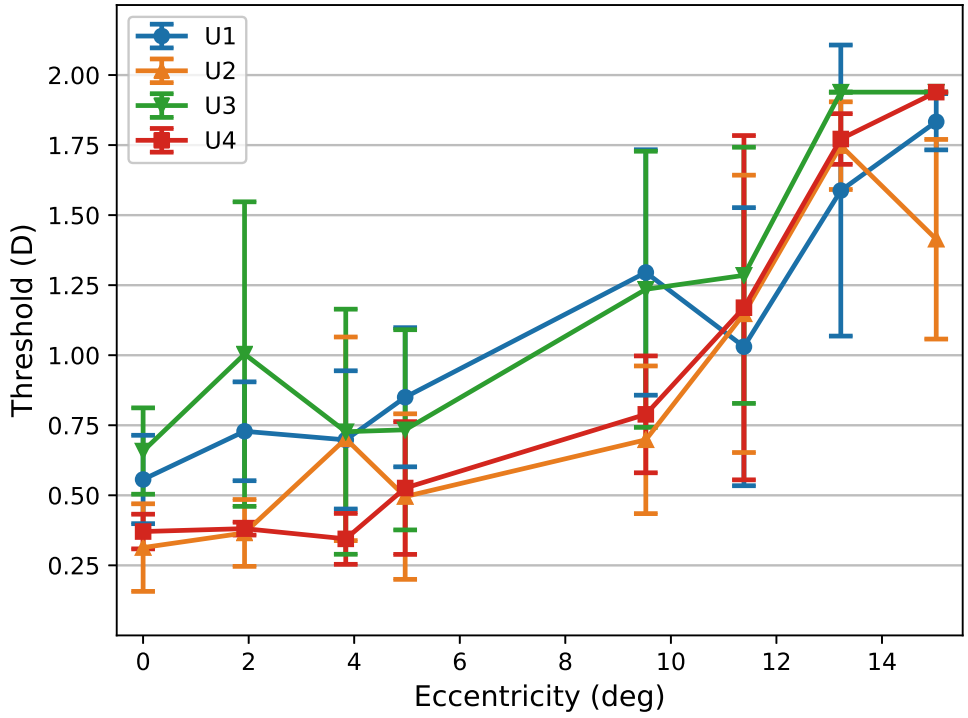


(a) *user* **(b)** *stimuli*

Figure 22: Depth perception studies (Section 4.1.2). *(a) shows the study setup. (b) shows a simulated retinal image from the light field stimuli. The green object is the fixation; the other two are the test targets.*



(a) *existence result*



(b) *thresholds result*

Figure 23: Depth perception study results. *(a) shows the study results of telling relative depth in fovea and periphery. Users show accurate perception in the fovea; yet in the periphery, the accuracy is near to random guess (50%). (b) shows the thresholds of depth disparity sensitivity with respect to eccentricity.*

4.2 Discussion

The blur perception thresholds from optical stimulus (Figure 21) show that some individuals retain high blur sensitivity as far as $15\,\mathrm{deg}$, with the minimum threshold down to $0.2D$. This finding rejects the straightforward idea of rendering only 2D images at far periphery; sufficient 4D sampling is still required.

However, it seems that the visual system can appreciate only a limited number of discrete rays generated by light field displays $-$ Figures 23a and 23b show the existence and consistency of decreased depth perception along eccentricities $\geq 6\,\mathrm{deg}$. This suggests that the thresholds measured using optical blur might be too conservative for light field displays.

The goal of foveating light field displays is faster rendering without degrading perceptual quality, especially depth cues. The different trends above suggest that foveated depth perception relates to both the human vision and the display systems. These motivate us to use the anatomically retinal receptor data ([72]), whose trend matches Figure 23b, to formally model all the retina-lens-display components in Section 5.4.

Despite the optical and biological factors above, our detection capability at a given eccentricity also relates to the stimuli's depth disparities, as observed from Figures 21 and 23b. This inspired us to extend the perceptual model for scene content adaptation in Section 5.4.

5 Foveation

A variety of applications such as virtual reality and immersive cinema require high image quality, low rendering latency, and consistent depth cues. 4D light field displays support focus accommodation, but are more costly to render than 2D images, resulting in higher latency. The human visual system can resolve higher spatial frequencies in the fovea than in the periphery. This property has been harnessed by recent 2D foveated rendering methods to reduce computation cost while maintaining perceptual quality. Inspired by this, we present foveated 4D light fields by investigating their effects on 3D depth perception. Based on our psychophysical experiments and theoretical analysis on visual and display bandwidths, we formulate a content-adaptive importance model in the 4D ray space. We verify our method by building a prototype light field display that can render only $16\% - 30\%$ rays without compromising perceptual quality.

5.1 Previous Work

A comfortable and immersive 3D experience requires displays with high quality, low latency, and consistent depth cues.

Depth perception and light field display Understanding and navigating 3D environments require accurate depth cues, which arise from multiple mechanisms including motion parallax, binocular vergence, and focus accommodation [73]. Conventional 2D desktop and stereoscopic displays lack proper focus cues and can cause vergence-accommodation conflict [74]. Although light field displays can support proper focal cue by 4D light rays [8, 31, 75, 76], they are considerably more costly to render or acquire than 2D images. Thus, they often lack sufficient speed or resolution for fully immersive VR applications which are sensitive to simulator sickness. Despite prior physiological studies in retinal blur and cell distributions [3, 72], it remains an open problem to build a perceptually accurate and quantitative model for fast content synthesis for light field displays. This project aims to address this challenge and answer the fundamental question: how should we sample a 4D light field to support focal cues with minimum cost and maximum quality?

(a) *user interface*

(b) *foveation left, focus far*

(c) *foveation right, focus far*

(d) *foveation right, focus near*

Figure 24: Foveated light field display and rendering. *(b), (c), (d) are our simulated retinal images under foveation with different tracked eye gazes (shown in green circles) and different focus planes. Specifically, (c) has the same gaze position but different focus plane from (d), and the same focus plane but different gaze position from (b). Our method traces only 25% of the light field rays while preserving perceptual quality.*

Foveated rendering The human visual system has much denser receptors (cones) and neurons (midget ganglion cells) near the fovea than the periphery. Foveated rendering harnesses this property to reduce computation cost without perceptual quality degradation in desktop displays [33] and VR HMDs [9]. The potential benefits of foveation for path tracing is surveyed by Koskela et al. [77]. However, foveation has not been explored in higher dimensional displays, such as for 4D light fields.

This work explores sampling/reconstruction algorithms and hardware requirements. The goal is to bridge 2D foveated rendering and 4D light field displays with perceptual preservation.

Light-field sampling Light field analysis in the spectral [78–81] or ray-space [82, 83] domain improves quality and performance of rendering [84–89] and acquisition [79, 90–93]. Prior work on light field rendering and reconstruction [84, 88, 94] focuses on the projected 2D images with distributed effects, for example, depth of field [89], motion blur [85], and soft shadows [86, 89]. However, foveating light field displays needs sparsely sampled 4D rays with sufficient fidelity for the observer to accommodate the scene content and integrate the retinal image.

Using gaze tracking, we augment traditional 4D light field sampling and rendering with two main components: visual foveation and accommodation. The former guides sampling to the retinal cells distribution; the latter allows adaptation to the scene content.

5.2 Overview

To understand the visual factors, we perform perceptual studies with both optical blur and our light field display prototype [7]. Driven by the study discoveries, we further analyze the whole light field system, including the display, the eye lens, and the eye retina, in both the primary and frequency domains in Section 5.3. Based on this perceptual model, we describe our 4D sampling and reconstruction methodology for foveated light field rendering in Section 5.4, and implementation details including hardware prototype and software system in Section 5.5. We validate our system via psychophysical studies and performance analysis in Section 5.6.

5.3 Analysis: Frequency Bounds

Light field displays require dense sampling from multiple viewpoints, which are orders of magnitude more expensive to render than traditional displays. Sheared filters with spatial-angular frequency bounds save samples for global illumination [85–87, 89]. However, image reconstruction from a 4D light field display is

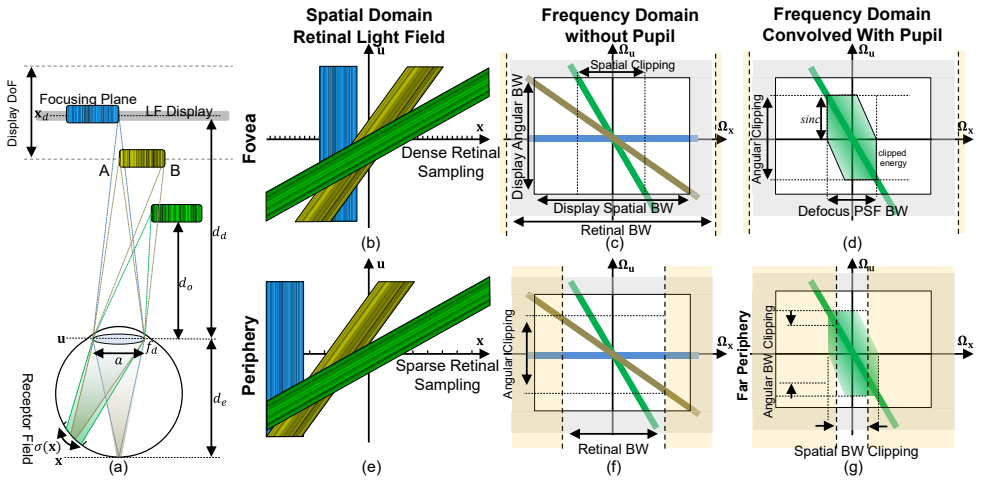


Figure 25: Light-field analysis in ray space and frequency domain. *The setup (a) of the eye focusing on the display has a foveal and a peripheral light fields shown in (b) and (e), and their frequency domain spectrum in (c) and (f) respectively. The perceivable light field is subject to spatial clipping due to the display bound (c) shown in retinal coordinates, angular clipping due to the lens bound (d), and spatial and angular clipping due to the retina bound (f). The final perceivable spectrum is obtained by aggregating all bounds (g): the narrower spatial retinal bound not only reduces the spatial bandwidth, but it also further lower the angular bandwidth from (d).*

automatic through and further bounded by human eyes. Thus, we derive spatial-angular frequency bounds in the realms of display, lens, and retina. The outcome of this analysis and the subsequent sampling strategy (Section 5.4.1) also answer the long standing question on the minimum number of rays required to support accommodation with a light field display.

In the ray space, we model the perceived retinal image $I(\mathbf{x})$ (Figure 25a) as an angular integration of the retinal light field $L(\mathbf{x}, \mathbf{u})$ (Figure 25b) across the pupil $\sqcap(\mathbf{u}/a)$. The corresponding frequency spectrum (Figure 25c, colored lines) is then obtained through Fourier slice theorem:

$$\begin{aligned} I(\mathbf{x}) &= \int L(\mathbf{x}, \mathbf{u}) \sqcap (\mathbf{u}/a) \mathrm{d}\mathbf{u} \\ \hat{I}(\omega_{\mathbf{x}}) &= \left(\hat{L} \star \hat{\sqcap} \right) (\omega_{\mathbf{x}}, \omega_{\mathbf{u}} = 0) \end{aligned}, \tag{28}$$

where $\hat{\cdot}$ denotes Fourier transform and $\star$ denotes convolution. When the eye has

focal length f and diameter d_e, the frequency domain slope of any out-of-focus object at depth d_o is

$$\frac{\omega_{\mathbf{u}}}{\omega_{\mathbf{x}}} \triangleq \hat{k}(d_o, f) = -d_e \left(\frac{1}{d_e} + \frac{1}{d_o} - \frac{1}{f} \right). \tag{29}$$

We approximate the spherical eyeball via a 2-plane parameterization, which suffices in many cases as the fovea is only within 5 degree and the periphery is blurred. A spherical parameterization [93] will be more accurate to model the retinal geometry and other phenomena, for example, Stiles-Crawford effect. Detailed derivations of Equations (28) and (29) and ray space analysis are shown by Huang et al. [37] and Section 5.9.1. Note that the slope $\hat{k}$ is linearly proportional to objects' diopter depths because both are inverses of metric depths.

Retina bound The spatial resolution of retina decreases with larger eccentricity primarily because the midget Retinal Ganglion Cell receptor field (mRGCf) increases dendritic field size [95] while maintaining a constant area sampling rate [96]. This inspires recent work [9, 33] in reducing the rendering cost via foveation. The visual acuity falls monotonically as the visual eccentricity grows, and the fall-off is known to follow the density of ganglion cells [97]. Watson [72] combined results from several studies to construct a model that predicts the receptive field density of midget ganglion cells as a function of retinal eccentricity $r = \sqrt{x^2 + y^2}$, for $(x, y) \in \mathbf{x}$ and the meridian type m:

$$\begin{aligned} \rho(r, m) = 2 \times \rho_{cone} \left(1 + \frac{r}{41.03} \right)^{-1} \\ \times \left[a_m \left(1 + \frac{r}{r_{2,m}} \right)^{-2} + (1 - a_m) \exp\left(-\frac{r}{r_{e,m}} \right) \right], \end{aligned} \tag{30}$$

where $\rho_{cone} = 14,804.6 \deg^{-2}$ is the density of cone cell at fovea and $a_m, r_{2,m}, r_{e,m}$ are all fitting constants along the four meridians of the visual field (details can be found in [72]). Figures 28a and 28b visualize the densities. In practice, we use the spacing

$$\sigma(\mathbf{x}) = \sigma(x, y) = \frac{1}{r} \sqrt{\frac{2}{\sqrt{3}} \left(\frac{x^2}{\rho(r, 1)} + \frac{y^2}{\rho(r, 2)} \right)} \tag{31}$$

to derive the retinal spatial bandwidth:

$$B^{retina}_{\omega_{\mathbf{x}}}(\mathbf{x}) = 1/(2\sigma(\mathbf{x})). \tag{32}$$

Figures 28c and 28d show corresponding sampling based on this bandwidth bound only. The corresponding angular bandwidth is obtained from the definition of $\hat{k}$ in Equation (29):

$$B^{retina}_{\omega_{\mathbf{u}}}(\mathbf{x}) = \hat{k}(d_o, f) B^{retina}_{\omega_{\mathbf{x}}}(\mathbf{x}). \tag{33}$$

The angular bound depends on both content depth and gaze eccentricity. The example in Figure 25f shows different angular bounds for objects at the same eccentricity.

Lens bound For an out-of-focus object, its perceivable frequency spectrum is governed by the energy contributed to the slicing axis $\omega_{\mathbf{u}} = 0$ in Equation (28) through convolution with the Fourier transformed pupil function $\hat{\sqcap}(\mathbf{u}/a) = \mathrm{sinc}(a\omega_{\mathbf{u}})$. The bounds are primarily limited by the pupil aperture a, and because $\mathrm{sinc}(\cdot)$ degrades rapidly after its first half cycle π, as shown in Figure 25d, we can derive the angular bandwidth $B^{lens}_{\omega_{\mathbf{u}}} = \pi/a$, and the corresponding spatial bandwidth is given by:

$$B^{lens}_{\omega_{\mathbf{x}}} = \begin{cases} \frac{\pi}{a\hat{k}(d_o,f)}, & \text{if } a > \frac{2\pi d_e \Delta \mathbf{x}_d}{\hat{k}(d_o,f) d_d} \\ \\ \frac{d_d}{2 d_e \Delta \mathbf{x}_d}, & \text{otherwise,} \end{cases} \tag{34}$$

where $\frac{d_e}{d_d}\Delta\mathbf{x}_d$ is the spatial sampling period of the light field display projected onto the retina, and it caps the spatial bandwidth by $1/\left(2\frac{d_e}{d_d}\Delta\mathbf{x}_d\right) = \frac{d_d}{2d_e\Delta\mathbf{x}_d}$ (the *otherwise* clause). The *if* clause has further reduced bound due to the object slope $\hat{k}(d_o, f)$.

Display bound Let $\Delta\mathbf{x}_d$ and $\Delta\mathbf{u}_d$ be the spatial and angular sampling periods of the display. With its angular bound $B^{display}_{\omega_{\mathbf{u}}} = 1/(2\Delta\mathbf{u}_d)$, Zwicker et al. [98] have shown a spatial bound $B^{display}_{\omega_{\mathbf{x}}}$ when an object's depth extends outside the depth of field of the display (Figure 25c); details are described in Section 5.9.2.

Overall bound The aforementioned bounds are aggregated into the smallest bandwidth among them:

$$B^{all}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}(\mathbf{x}) = \min\left(B^{retina}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}, B^{lens}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}, B^{display}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}\right)(\mathbf{x}), \tag{35}$$

An example is shown in Figures 25a and 25g.

How many rays do we need? It has been asked for a decade that how many rays entering the pupil, that is, the angular sampling rate, are needed for a light field display to support proper focus cue. As we have studied and derived, the display, the optics of the eye, and the anatomy of the retina all affect the final perceivable image. Based on the discoveries, we present a closed-form and spatially-varying ray sampling strategy in Section 5.4.

5.4 Method: Sampling and Rendering

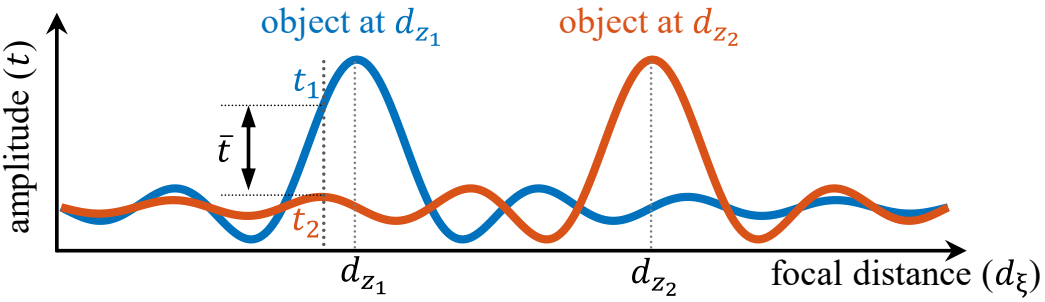


Figure 26: Sampling strategies illustration. *X-axis represents the accommodative depth* d_ζ. *Y-axis shows the amplitude* t *from Equation* (37). *Varying objects depths demonstrate different amplitude distribution w.r.t* d_ζ. *The differential amplitude* $\bar{t}$ *in Equation* (38) *is the distance between intersections.*

The bandwidth bounds in Section 5.3 include optical and retinal components. However, variations in scene depth content [7], the eye's focus and movement ([3, 99]), and occlusions [100] also decide our depth perception. Considering those additional factors, we extend the bounds in Equation (35) for an importance-based model for sampling and rendering. As illustrated in Figure 26, we consider the perceived amplitude difference among objects ($\bar{t}$) as the depth stimulus strength. Based on this, we derive an importance value W for each light ray $(\mathbf{x}, \mathbf{u})$ with regard to the static range and dynamic movements of accommodative depth d_ζ. This importance distributes the ray budget for the final shading and filtering.

5.4.1 Content-Adaptive Light Field Sampling

To formally analyze the increased importance due to occlusion, consider two objects at distances d_{z_1} and d_{z_2} to the eye and are visible within a small window centered on a light ray $(\mathbf{x}, \mathbf{u})$. In the frequency domain, their retinal light field spectra have slopes $\hat{k}(d_{z_1}, f_\zeta)$ and $\hat{k}(d_{z_2}, f_\zeta)$ (Equation (29)) with a time-varying focal length of the eye f_ζ. When they are out-of-focus, their perceivable bandwidth with respect to the focus distance[1]

$$d_\zeta = \left(\frac{1}{f_\zeta} - \frac{1}{d_e}\right)^{-1} = \frac{f_\zeta d_e}{d_e - f_\zeta} \tag{36}$$

to the eye is equal to the contribution of amplitude spreading toward the slicing axis $\omega_\mathbf{u} = 0$, and is given by

$$t(d_{z_i}, d_\zeta, \omega_\mathbf{x}) = \left\| \hat{s}_i \left(-\frac{d_e}{d_{z_i}} \omega_\mathbf{x} \right) \right\| \operatorname{sinc} \left(a \omega_\mathbf{x} \hat{k} \left(d_{z_i}, f_\zeta \right) \right), \tag{37}$$

where $\|\hat{s}\|$ is the amplitude of the surface texture in the frequency domain. Please refer to Huang et al. [37] and Section 5.9.6 for detailed derivations. In monocular vision, the eye perceives depths through the differences in the defocus blur. Thus, given the constant focusing distance d_ζ, we consider their differences in the perceivable signal amplitudes:

$$\bar{t}(d_{z_1}, d_{z_2}, d_\zeta, \omega_\mathbf{x}) = \left\| t(d_{z_1}, d_\zeta, \omega_\mathbf{x}) - t(d_{z_2}, d_\zeta, \omega_\mathbf{x}) \right\|. \tag{38}$$

Static sampling Following our blur and depth perception studies [7], and the display-eye bandwidth discussions (Section 5.3), Equation (38) presents an analytical modelling for defocus blur with a constant focusing distance and two objects, as visualized in Figure 26. We consider all the visible objects within a ray and compute the corresponding importance indicator for sampling:

$$\begin{aligned} w_s(d_\zeta) &= \sum_{\forall i,j \in objects}^{i \neq j} \int_{\Omega_x} \bar{t} \left(d_{z_i}, d_{z_j}, d_\zeta, \omega_\mathbf{x} \right) \mathrm{d}\omega_\mathbf{x} \\ &\propto \int_{\Omega_x} \bar{t} \left(d_z^-, d_z^+, d_\zeta, \omega_\mathbf{x} \right) \mathrm{d}\omega_\mathbf{x}, \end{aligned} \tag{39}$$

[1] d_ζ is focal distance, f_ζ is focal length, as illustrated in Figure 27.

where $[d_z^- = \min_{\forall i} d_{z_i}, d_z^+ = \max_{\forall i} d_{z_i}]$ is the scene's local depth range around the ray. The above formulation requires the knowledge of focal distance d_ζ, which is not directly available due to lack of accommodation tracking technologies. We address this limitation by integrating d_ζ over the estimated accommodation range $[d_\zeta^-, d_\zeta^+]$ for the final importance estimation in Equation (41). The real-time acquisition of $d_\zeta^\pm$ and $d_z^\pm$ are described in Section 5.5.

Dynamic sampling The static weighting above considers a fixed d_ζ. However, accommodation can also be guided by the modulation of retinal images as the eye changes its focal distance (e.g., through micro fluctuation [99]). These motivate us to consider a dynamic factor that reflects a changing d_ζ:

$$w_d(d_\zeta) = \int_{\Omega_x} \frac{\partial \bar{t}\left(d_z^-, d_z^+, d_\zeta, \omega_{\mathbf{x}}\right)}{\partial d_\zeta} \mathrm{d}\omega_{\mathbf{x}}. \tag{40}$$

Figure 27 shows the matching trend between normalized $w_d(d_\zeta)$ and prior vision science discovery from Watson and Ahumada [3] that the strongest blur discrimination occurs when the accommodation depth (d_ζ) lies slightly off-center to object depths ($d_z^\pm$).

Overall sampling Combining the above stimuli strengths modeled with scene content and accommodation preference, we have the importance $w_d(d_\zeta)w_s(d_\zeta)$ for a specific focal distance d_ζ. To fully construct the importance for a light ray $(\mathbf{x}, \mathbf{u})$, we consider its effective local amplitude differences by integrating over the focal distance range $[d_\zeta^-, d_\zeta^+]$. We estimate this range as the min-max depths in fovea since people usually observe and focus on objects within this area. To further accelerate the calculation, we transform each integration to a uniform coordinate frame (via the operator η below):

$$\begin{aligned} W(\mathbf{x}, \mathbf{u}) &= \int_{d_\zeta^-}^{d_\zeta^+} w_d(d_\zeta) w_s(d_\zeta) \mathrm{d}d_\zeta \\ &\overset{\eta}{=} \int\int w'_d\left(\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}}\right) w'_s\left(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}\right) \mathrm{d}\omega'_{\mathbf{x}} \mathrm{d}\omega'_{\mathbf{u}}, \end{aligned} \tag{41}$$

where $(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}) = \eta(d_\zeta, \omega_{\mathbf{x}}, \omega_{\mathbf{u}})$ is the transformed frequency coordinate, and $\{w'_s, w'_d\}$ are the pointwise importance functions in the new frame; details are

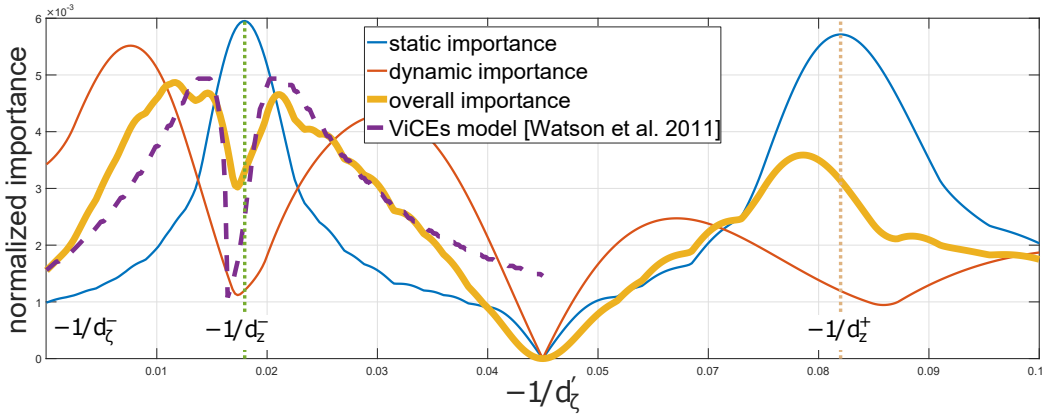


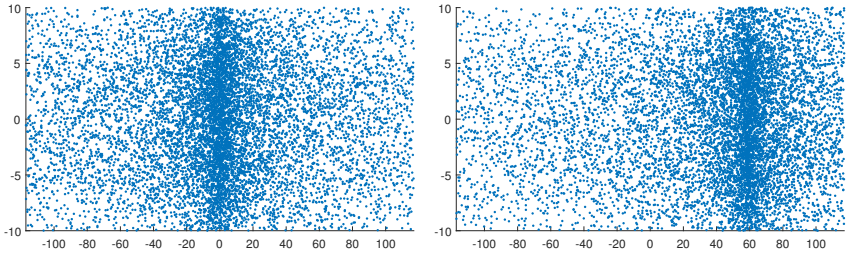
Figure 27: Importance values and the model from [3]. *The three solid curves plot normalized values of Equations* (39) *to* (41) *in transformed coordinate (Section 5.9.3). The dashed curve shows the trend of depth perception of the object at depth* $d_z^- = 4D$ *from ViCEs prediction model [3] by assuming its inversed detectable threshold to be the importance. The x-axis represents different accommodation* d'_ζ *within the range of* d_ζ^- *and object at depth* d_z^+*. Because the ViCEs model considers only one of those two objects due to symmetry, its plot has the x-axis range between* d_ζ^- *and* $\frac{d_z^- + d_z^+}{2}$*. Coordinates of* d'_ζ *are transformed as* $\frac{-1}{d'_\zeta}$ *for easier visualization. Symbols are illustrated in Figure 29.*

derived and discussed in Section 5.9.3. The integrating ranges in Equation (41) are bounded by the frequency bandwidth $B^{all}_{\{\omega_\mathbf{x},\omega_\mathbf{u}\}}$ in Equation (35), and the range of focal length and distance:

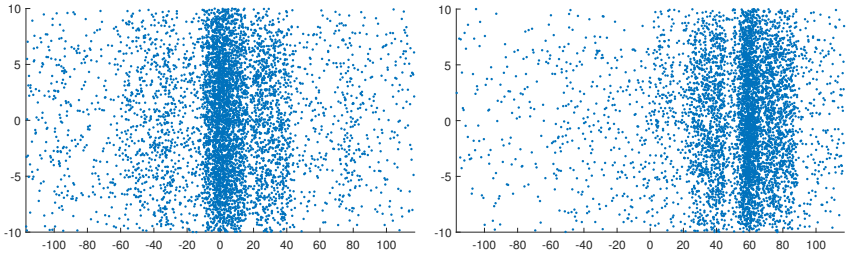
$$\begin{aligned}(\omega_\mathbf{x},\omega_\mathbf{u}) &\in [-B^{all}_{\omega_\mathbf{x}}(\mathbf{x}), B^{all}_{\omega_\mathbf{x}}(\mathbf{x})] \times [-B^{all}_{\omega_\mathbf{u}}(\mathbf{x}), B^{all}_{\omega_\mathbf{u}}(\mathbf{x})] \\ \frac{\omega_\mathbf{u}}{\omega_\mathbf{x}} &\in [\hat{k}(d_\zeta^-, f_\zeta^-), \hat{k}(d_\zeta^+, f_\zeta^-)]. \end{aligned} \tag{42}$$

This analytical importance function can be computed in closed form to allow real-time performance, as is shown in Section 5.9.6. It guides spatially-varying and perceptually-matching ray allocations given a specified rendering budget. As visualized in Figures 26 and 27, our min-max estimation will only increase the numbers of samples, thus being more conservative. In Section 5.9.4, we also present the minimum budget required given a display-viewer setup.

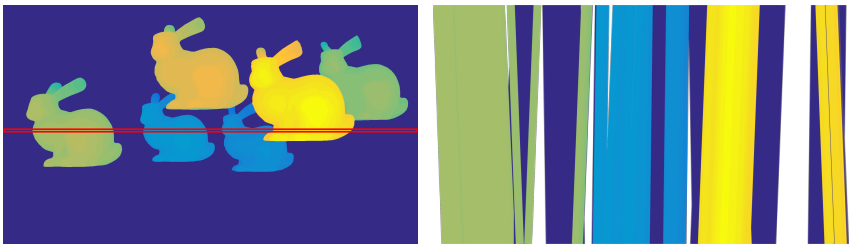
(a) *retina projection display center* **(b)** *retina projection display side*



(c) *ray space sampling for (a)* **(d)** *ray space sampling for (b)*



(e) *content adapt sampling from (c)* **(f)** *content adapt sampling from (d)*



(g) *scene depth* **(h)** *flatland for the red line in (g)*

Figure 28: Spatial-angular content adaptive sampling. *(a) and (b) show the retinal ganglion density (Equation* (30)*) projected on the display when the gaze is at the center or side of the display. (c) and (d) show the corresponding ray space sampling for (a) and (b). Based on (c) and (d), (e) and (f) further adapt to the content shown in (g) and (h). The flatland visualizations in (c), (d), (e), (f), and (h) are in the display space with* mm *as units in both axes.*

5.4.2 Sparse Sampling and Filtering for Rendering

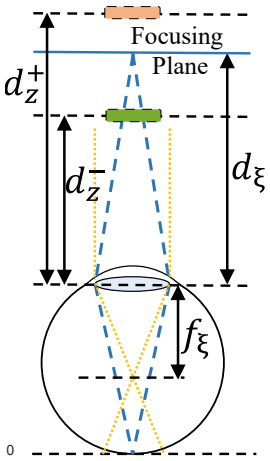


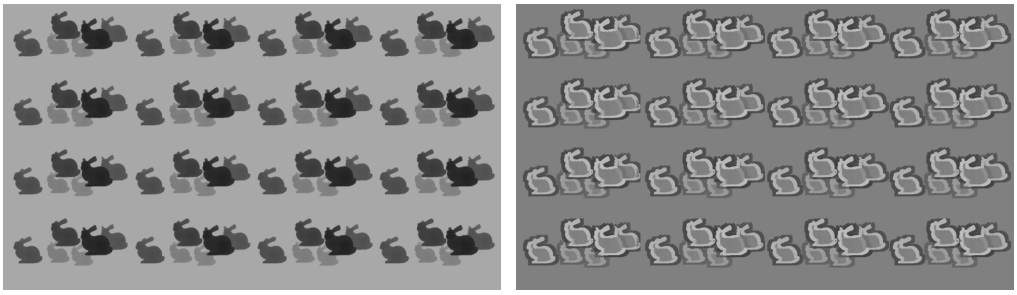
Figure 29: Symbols for Figure 27.

We perform a two-stage GPU-based **sampling** to realize the importance model above, as visualized in Figure 28. To compute preliminary saving (Figures 28c and 28d) without expensive global Fourier transform, we first estimate each local ray region's maximum sample number s_{el} (Section 5.9.4) by distributing the total budget with retina bounds $B^{retina}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}(\mathbf{x})$ to consider eccentricity effect. We then compute, for each ray, its aggregate bounds $B^{all}_{\{\omega_{\mathbf{x}},\omega_{\mathbf{u}}\}}$ (Equation (35)) to delineate the domain (Equation (42)) for the importance value $W(\mathbf{x},\mathbf{u})$ in Equation (41). We multiply s_{el} with W/ξ to finalize the sample count for each ray (Figures 28e and 28f). ξ is a global ratio to rescale W into $[0,1]$, with $\xi = 320$ based on our specific hardware setup and experiments to balance between performance and perceptual quality. ξ can be further increased for stronger savings, but more thorough evaluation may be needed. To avoid zero samples for flat regions, we clamp the ratio W/ξ to be within $[0.3,1]$. The min clamping value 0.3 can be further reduced with higher resolution displays (e.g., 4K instead of 2K).

The sparsely sampled ray set is **filtered** for rendering a light field display with uniformly spaced pixels. We implement a separable 4D Gaussian radial basis function for the sparse reconstruction and handle occlusions using the coarse depth map (Figure 30); details are shown in Section 5.9.5. Finally, similar to [9], a contrast-preserving filter is applied to improve quality.

5.5 Implementation

Depth disparity estimation In each frame we render a multi-view low spatial resolution (500×300) depth mipmap, as shown in Figure 30a, to estimate the local depth variations. Specifically, depending on the specific scene complexity, we render no more than 4×4 depth maps using simultaneous multi-viewport projection supported by modern GPUs. From this multi-view depth mipmap, we find the local minimum and maximum depth for each coarse pixel by performing a mix-max comparison around the local neighborhood and pyramid layers, as show



(a) *low res zbuffer* **(b)** *low res analysis*

Figure 30: Depth disparity estimation of local regions. *(a): Depth buffer from multiview projection. (b): Real-time depth disparity analysis of local regions; with brighter colors representing larger disparities.*

in Figure 30b. Combining the two maps using bilinear interpolation, we obtain the values of $d_\zeta^\pm$ and $d_z^\pm$ to compute Equation (41) for any ray $(\mathbf{x}, \mathbf{u})$.

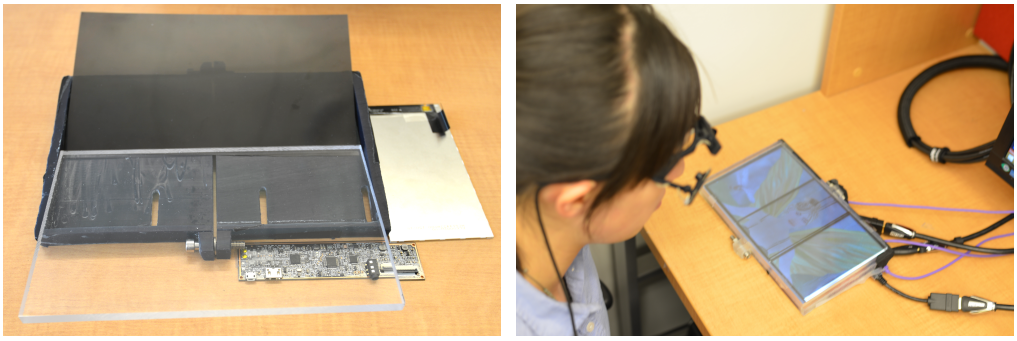
Ray-tracing We implement our system using the NVIDIA OptiX ray tracer. For comparison, we also implement two full-resolution light field rendering techniques by ray tracing [31] and rasterization [8].

The foveated rendering pipeline requires asynchronous computation of importance sampling. Thus, we separate the rendering into two stages similar to the decoupled shading [101]: we first create a queue of rays to be shaded, and then use the scheduler to processes the shading. Similar to the foveated rasterization [9], we also suffer performance penalty without dedicated hardware scheduler which supports coarse pixel shading. However, our method still shows performance gains in both frame rates and number of shaded rays; see Figure 40.

Hardware To validate the foveated light field rendering, the prototype hardware needs to offer a high spatial/angular resolution, a wide depth of field , and a wide field of view to separate foveal and peripheral regions. We build a parallax-barrier based light field display by tiling three 5.98-inch 2560×1440 panels (part number TF60006A) from Topfoison. The parallax-barrier at $9.5mm$ from the panels is printed with $300\mu m$ pitch size using a laser photoplotter; its pinhole aperture is $120\mu m$ to avoid diffraction. The final light field display has 579×333 hardware

spatial resolution at 10-inch diagonal size and 8×8 views angular resolution (3.2 views/degree), larger than the 5×5 angular resolution in, [8], which can already support proper accommodation. The components and the interfaces are shown in Figure 31. Assuming an eye with $6mm$ pupil aperture viewing the display from $30cm$ away, we ensure 10 rays/pixel entering the eye to support accommodation. The renderer is driven by a PC with an 2.0GHz 8-core CPU with 56GB of RAM, and an NVIDIA GTX 1080 graphics card. Example elemental image using our foveated ray tracing (Section 5.4) and the display hardware can be found in Figure 32.

We augment the light field display with a PupilLab [102] eye tracker. The head-mounted tracker offers real-time streaming of gaze positions in the display space. We drive the tracker with a laptop. The foveal accommodation range $[d_\varsigma^-, d_\varsigma^+]$ in Equation (42) are obtained by combining the eye-tracked gaze position and a ray propagation from the eye to the gaze.



(a) *components* **(b)** *system setup*

Figure 31: Our hardware design and system setup. *(a) shows components to build our light field display in Section 5.5. (b) shows our system setup: a user wearing glass-style eye tracker watches the display.*

5.6 Evaluation

For perceptual and performance evaluation, we choose 11 scenes with different depth distribution, geometric complexity, and field of view. Figures 24 and 40 show simulated renderings while Figure 33 shows captured results.

Figure 32: A foveated light field elemental image from the framebuffer of our prototype display. *The 4D light field is generated through propagating rays from each pixel to its corresponding pinhole. The tracked gaze position is at the face of the fairy. Please zoom-in for details.*

5.6.1 Perceptual quality

We conducted a user study to evaluate the effectiveness and quality of our method, by comparing with full-resolution and uniformly down-sampled light fields with the same number of rays as our method. Our goal is to show that foveated light fields achieve the quality of former with the performance of the latter.

Setup The experimental setup consisted of our prototype light field display, a head-mounted eye tracker [102], and machines (Section 5.5) that rendered and drove the system. We used a $12mm \times 12mm$ eye box at $0.3m$ from the display.

Stimulus The stimulus was the fairy scene. Objects contain both high and low spatial frequency textures. The light field of the stimulus was generated using one of the three methods: full resolution, foveated, and uniformly downsampled. The full resolution condition sampled all the rays represented by the hardware

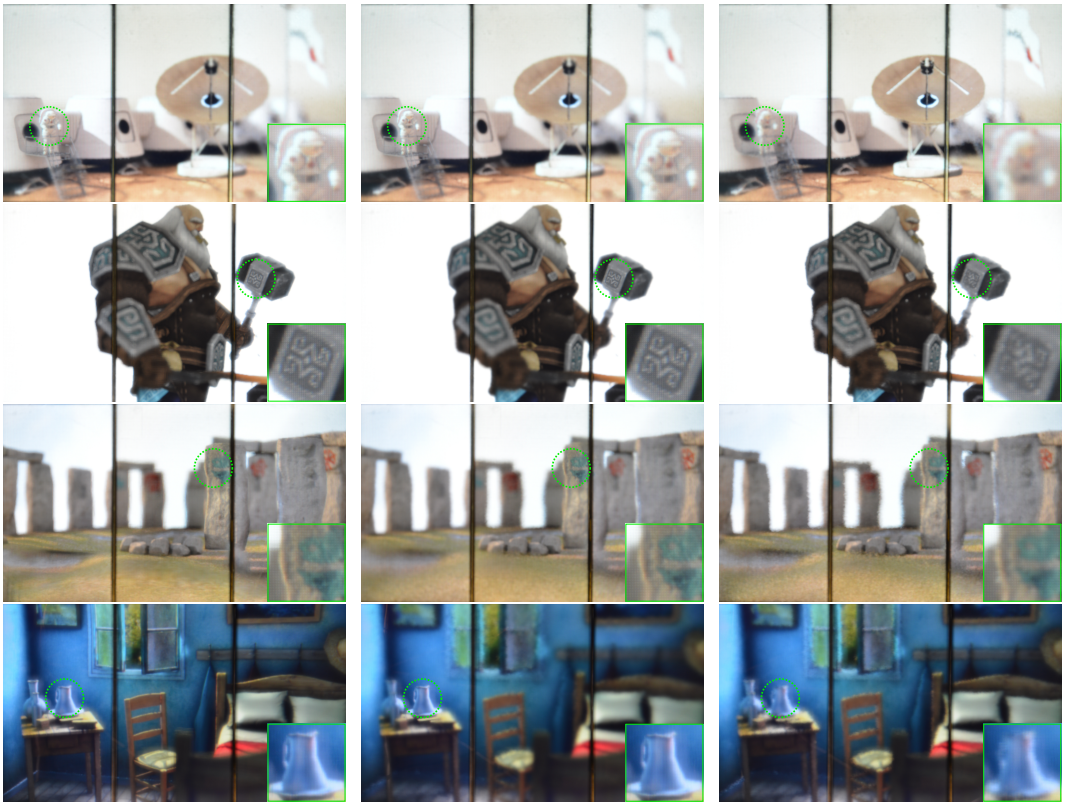
(579×333 spatial and 6×6 angular given the eyebox size). Foveated condition used our framework in Section 5.4, resulting in 24.8% samples (Table 3) compared with full resolution. Uniformly downsampled condition had the same number of rays as the foveated one but uniformly distributed the samples across retina.

Task Subjects examined and memorized details of the full resolution stimulus before the beginning of the experiment. During each trial, the display presented a stimulus rendered using one of the three methods for 4 seconds. Subjects were instructed to gaze at the fairy's head to avoid big saccades (fast and ballistic eye movements) and choose on keyboard about whether the stimulus looked the same as the examined full resolution stimulus. The entire experiment consisted of 42 trials, 14 per each rendering method. The order of all trials was randomized. Similar to previous studies on foveated effects ([9,103]), we inserted blank frames between trials. 14 subjects participated in the experiment (4 females and 10 males, aged 27 to 35). All subjects had normal or corrected-to-normal visual acuity. None of the subjects were aware of the experimental hypothesis or number of rendering methods.

Table 2: User study results. *The values are number of trials (out of 14) where subjects did not notice artifacts. Some subjects reported visible artifacts even in full-resolution condition, reflecting individual differences in criteria. The difference in perceived image quality was significant between full-resolution vs. uniform and foveated vs. uniform ($p < 0.0001$), but not significant between full-resolution vs. foveated ($p = 0.67$).*

case \ user	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
full resolution	12	14	6	12	14	13	14	12	14	13	7	14	14	13
foveated	14	14	6	7	13	13	14	7	10	14	9	14	14	12
uniform	4	5	0	0	2	0	0	0	0	8	0	0	0	4

Result Table 2 shows the number of trials where subjects reported that the stimulus looked the same as full resolution. A one-way within-subjects ANOVA showed that the effect of rendering method is significant ($F_{(2,26)} = 121.1, p < 0.0001$). Note that the difference in perceived image quality was significant between full-resolution vs. uniform and foveated vs. uniform ($p < 0.0001$, paired t-test with Bonferroni correction), but not foveated vs. full-resolution ($p = 0.67$).



(a) *full resolution* **(b)** *our foveated display* **(c)** *uniform down-sampling*

Figure 33: Photograph results from our prototype tiled display with 3 panels. *Our foveated results in (b) have similar quality to full-resolution rendering in (a), and higher quality than uniform sampling with the same number of rays in (c). Because uniform sampling does not consider either retinal receptor distribution or scene content, it introduces blur in fovea and aliasing near occlusion boundaries. The tracked gaze positions are marked in green circles with insets for zoom-in. All captured results are from our prototype (gamma correction enabled) in Figure 31 by a Nikon D800 DSLR camera with a 16-35mm f/4G lens. Corresponding retinal image simulations are available in the supplementary material. From top to bottom: Mars, craftsman, Stonehenge, van Gogh.*

The experimental results demonstrate that our framework lowers sampling rate without degrading perceived image quality. Figures 24 and 33 show more quality comparisons. Please refer to our supplementary video for live capture of a user

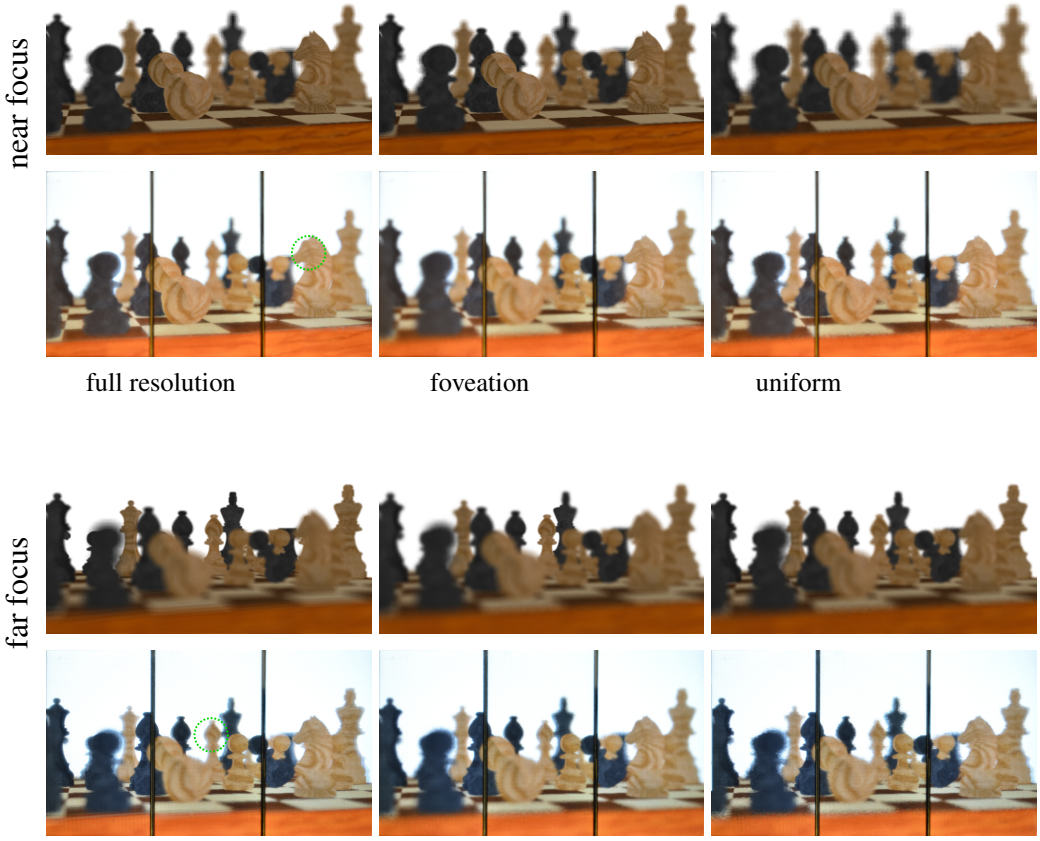


Figure 34: Chess scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*

interacting with our prototype display.

Figures 34 to 39 show extra simulated/captured images with near/far focus comparing full-resolution/foveated/uniform light fields for various scenes in our paper. Green circles on captured full-resolution images show tracked user gaze.

Figure 35: Fairy scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*

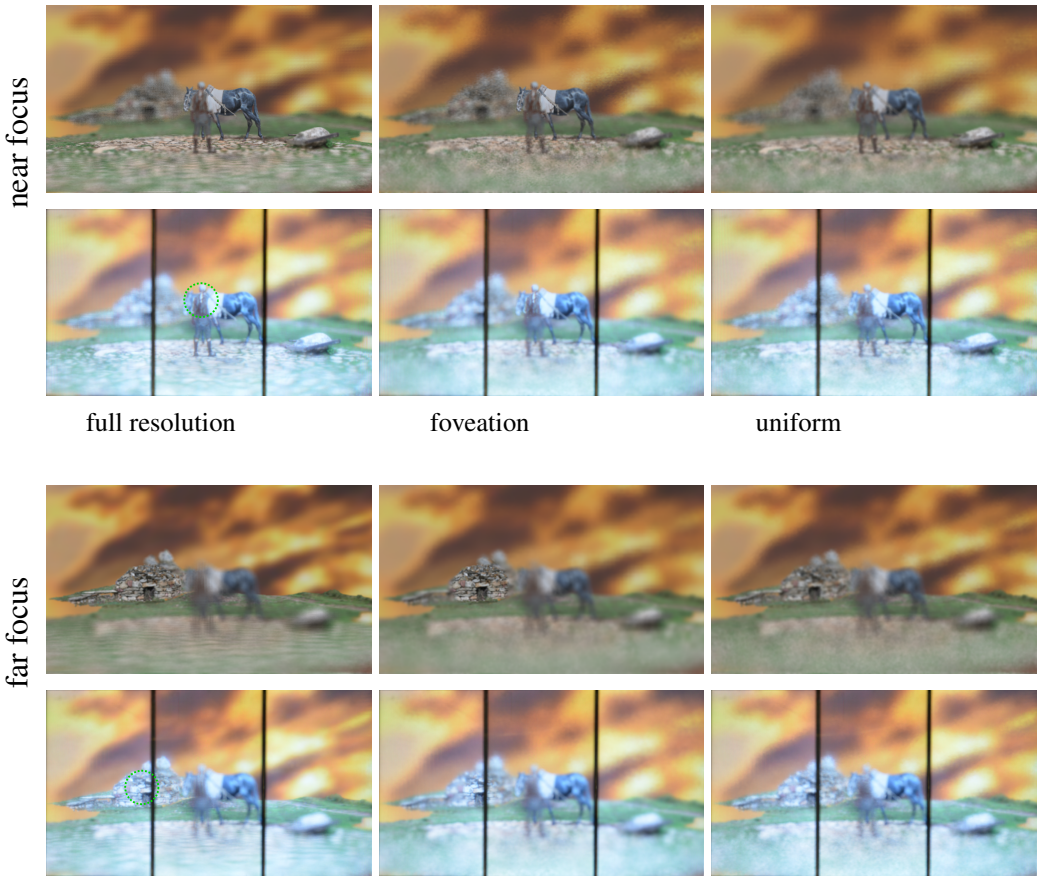


Figure 36: Farm scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*

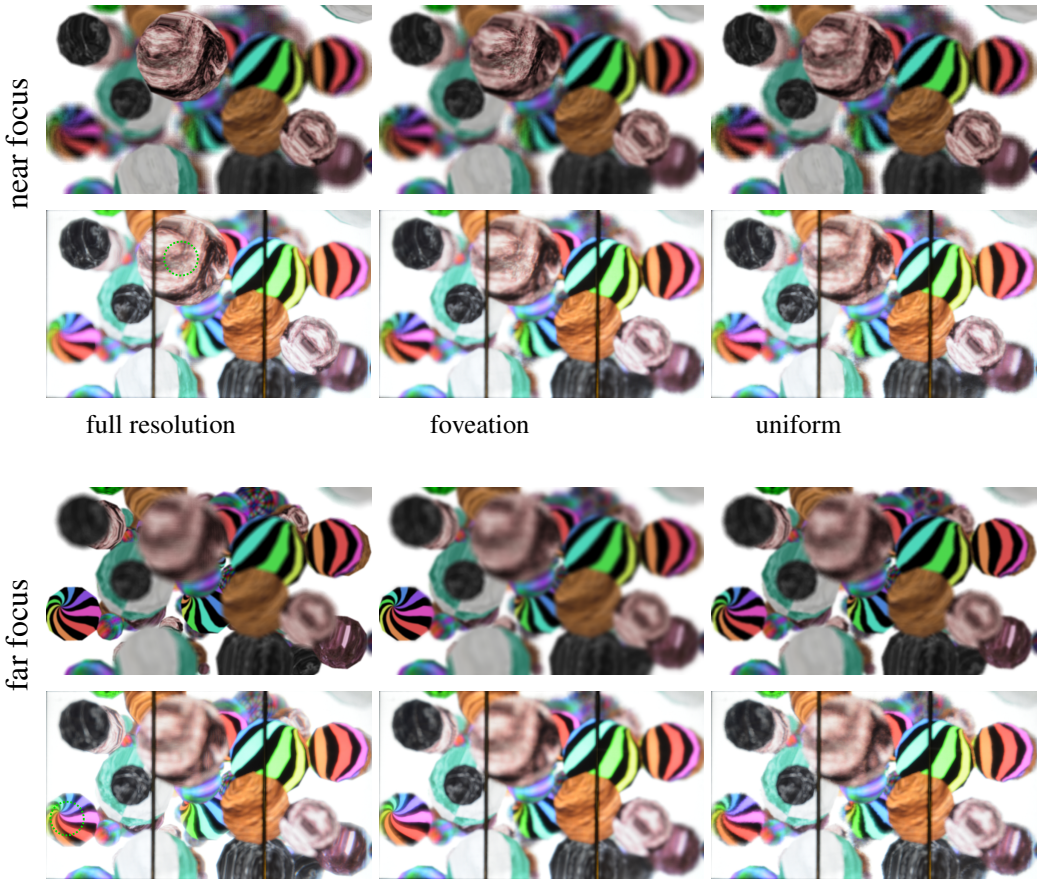


Figure 37: Marbles scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*

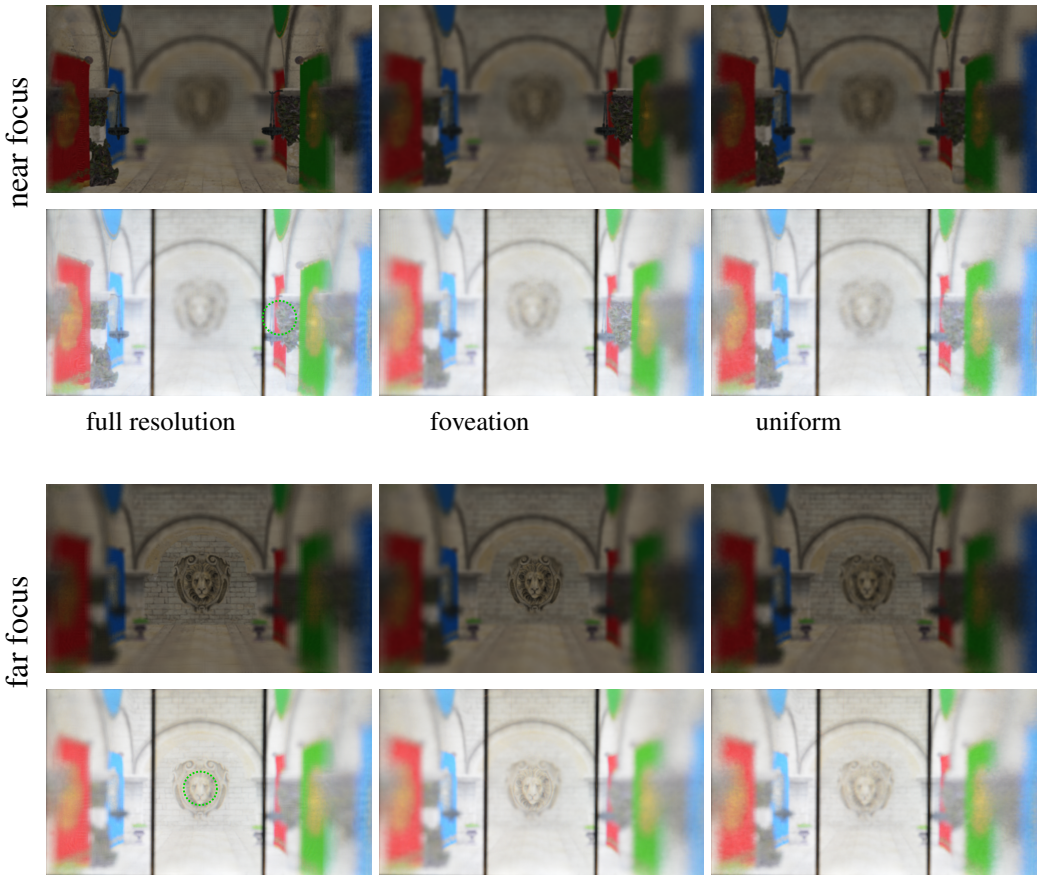
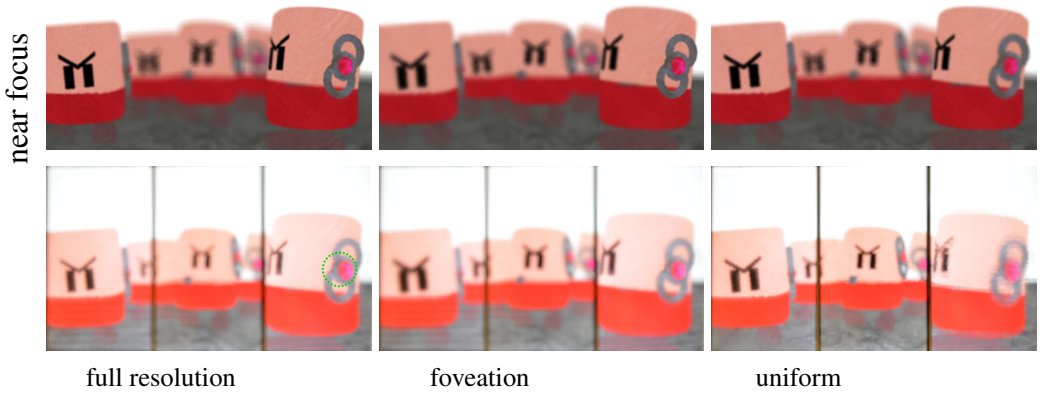


Figure 38: Sponza scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*



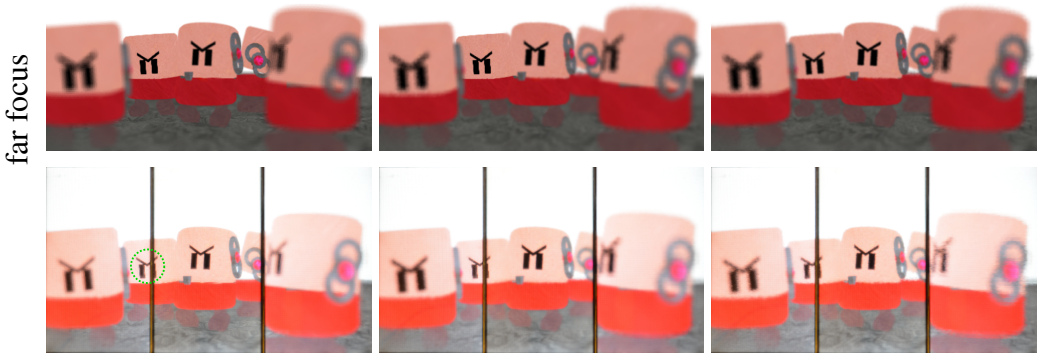


Figure 39: Toaster scene. *From top to bottom: near/far focus × simulation/capture results. From left to right: full-resolution, foveation, uniform sampling.*

5.6.2 Performance

Table 3: Ratio of number of traced rays in foveaton relative to full resolution.

plane	fairy	Mars	Sponza	toaster	farm
16.42%	24.80%	27.20%	29.38%	25.78%	24.69%
craftsman	marbles	Stonehenge	van Gogh	Viking	chess
24.67%	28.6%	21.59%	18.57%	24.59%	26.96%

Table 3 shows the ratio of the minimal number of traced light field rays with foveation (as computed in Section 5.9.4) compared with full resolution rendering. Since our method is content-adaptive, the saving in sampling and ray tracing is related to the scene complexities. One extreme scene is a flat plane, in which the ratio is only 16.42%. Our most challenging case is Crytek Sponza containing large variation in depth along the wall; the ratio increases to 29.38%, but the overall time performance is still $2\times$ faster than Huang et al. [8], as shown in Figure 40. Compared to the recent 2D foveated rendering method [9], our 4D light field foveation saves more pixel computation (up to 80%+ vs. up to 70%). Note that the method presented by Patney et al. [9] is constrained by GPU design thus only offer theoretical saving rather than actual performance (frame rates) benefit. Our system demonstrates actual performance gain with modern GPUs.

5.7 Discussion

Real-time foveated light fields involve multiple disciplines: display, rendering, content analysis, and human perception. Each component contains challenging open problems. We have proposed a starting point for this broad topic in which industry and consumers are gaining significant interests. Our current method and implementation still depend on the perceptual diversities of the observers [7], the precisions of trackers, and the capabilities of the GPUs.

Perception Our psychophysical data and perceptual model can benefit general foveated rendering goals focusing on accommodative depth perception, but other individual factors, including stereoscopic depth [104], high-order refractive aberrations, pupil size, eye dominance, prismatic deficiencies, contrast/color sensitivities, etc., may also influence light field perception. Thus, the saving can be

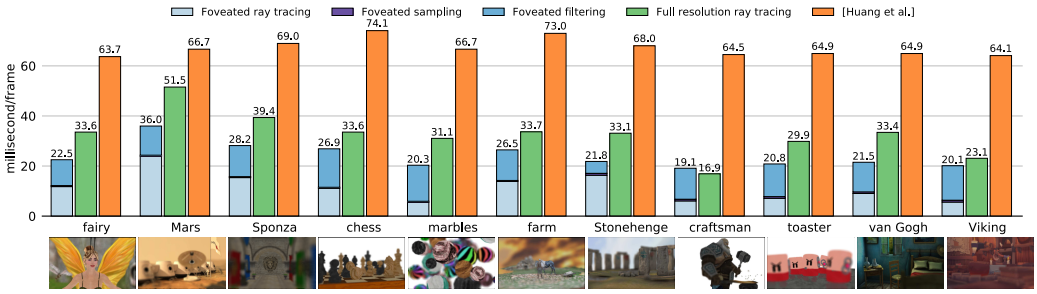


Figure 40: Performance comparison and breakdown. *Performance comparison with full resolution ray tracing [31] and rasterization [8]. Y-axis is the time consumption per frame measured in million-seconds. We also break down the timing for our method into the main components: sampling, ray tracing, and post-filtering. By sampling much less rays (Table 3), our method demonstrates lower overall computation costs, in particular the ray tracing part compared with full resolution ray tracing. Scene courtesies of Ingo Wald, admone, Crytek, Olexandr Zymohliad, Andrew Kensler, Raúl Balsera Moraño, ruslans3d, olmopotums, Andrew Kensler, rusland3d and nigelgoh respectively.*

conservative by using the bounds from the anatomical structure. Fully immersive VR/AR applications may require identification of thresholds at eccentricities wider than the $15\,\mathrm{deg}$ in our perceptual experiments. These factors are worth study as potential future works but beyond a single paper which first explores foveated light fields.

Tracking We discouraged users from making big saccades [7], but saccadic movement is known to help improve depth perception. While our entire system latency (tracker-renderer-display) is shorter than the accommodative reaction time, it is still longer than saccade-proof ($< 60ms$ [105]). Enlarging foveal area balances the system latency, but it affects the accuracy of the psychophysical data which derives and validates our methods. However, we believe the development of fast eye tracking and rendering hardware can help future foveated displays.

GPUs Rendering light field using ray-tracing might not be the optimal because modern GPUs are originally designed for rasterization. For the latter, further performance improvement can be achieved with future hardware supporting content

adaptive shading [106]. Our current implementation adds overhead on the post-filtering process (Figure 40), but similar to Heide et al. [107], integrating the rendering to a compressive display hardware could deliver better performance and image quality.

Scene Although we have analyzed the bandwidth bounds for Lambertian objects, highly specular 4D surfaces, (semi)transparent objects and high-frequent objects, need further examination on the extended area by the BRDF/BTDF bounds. The occlusion effect is not analyzed in our frequency analysis, so we can only address them in the spatial domain through importance sampling; insight from sheared filter in light transport [108] may contribute to this area. Our analysis and implementation do not consider the temporal dimension: sampling for temporal anti-aliasing across the retina [109] is a potential future direction.

5.8 Conclusion

Light field displays resolve the vergence-accommodation conflict that causes eye-strain and double vision, and improve 3D perception even for monocular vision. However, 4D light fields incur heavier rendering workload than 2D images. Inspired by the vision of Egan [110], we address this challenge by conducting content-aware physiological studies, deriving a perceptual model, and designing a real-time foveated 4D light field rendering and display system. Our prototype system offers both theoretical and actual performance gain with current GPUs (Section 5.6.2) and preserves perceptual quality when the visual system automatically reconstructs retinal images (Section 5.6.1).

Across the retinal eccentricity, going from the anatomical receptor distribution, spatial acuity, blur sensitivity, to the depth perception, is a long path. Each individual connection is a long standing research topic in the community. By analyzing the entire optical process from display to retina, our method guides an optimized allocation strategy given hardware budget and user input. It also suggests the minimum sampling required to provide proper accommodation. For the future, we envision 3D display technologies such as digital hologram for near eye display or vari-/multi-focal display can also benefit from foveated light fields.

5.9 Appendices

5.9.1 Ray Space Analysis

We first consider an observer focusing on a light field display at a distance $d_d = (d_e f_d)/(d_e - f_d)$ where f_d is the focal length of the eye when focusing on the display and d_e is the diameter of the eyeball, as shown in Figure 25a. The display light field L_d propagates along the free space and is refracted by the eye lens, and the retina receives an image I by integrating the retinal light field L along the angular dimension $\mathbf{u}$ parameterized at the pupil:

$$\begin{aligned} I(\mathbf{x}) &= \int L(\mathbf{x}, \mathbf{u}) \sqcap (\mathbf{u}/a) \mathrm{d}\mathbf{u} \\ &= \int L_d(\phi(\mathbf{x}, \mathbf{u}), \mathbf{u}) \sqcap (\mathbf{u}/a) \mathrm{d}\mathbf{u}, \end{aligned} \tag{43}$$

where a is the pupil aperture, $\sqcap(\cdot)$ is the rectangular function, and ϕ maps the intersection of a retinal light ray $(\mathbf{x}, \mathbf{u})$ with the display spatial point $\mathbf{x}_d$:

$$\begin{aligned} \mathbf{x}_d = \phi(\mathbf{x}, \mathbf{u}) &= -\frac{d_d}{d_e}\mathbf{x} + d_d \kappa(d_d, f_d)\mathbf{u}, \\ \kappa(d, f) &= \left(\frac{1}{d_e} - \frac{1}{f} + \frac{1}{d}\right). \end{aligned} \tag{44}$$

For an out-of-focus virtual object being presented at a distance $d_o \neq d_d$ to the eye, we can obtain its corresponding retinal light field through the inverse mapping of Equation (44), with slope

$$k(d_o, f_d) = (d_e \kappa(d_o, f_d))^{-1} \tag{45}$$

in the flatland diagram, as shown in Figure 25b. Since we integrate all rays over the pupil to obtain the retinal image in Equation (43), the image is blurred by a retinal Circle-of-Confusion (CoC) of diameter

$$CoC = \frac{a}{k(d_o, f_d)} = a d_e \kappa(d_o, f_d). \tag{46}$$

In the case of an out-of-focus object, intuitively we can sample it at frequency inversely proportional to the circle-of-confusion size. Similarly, inspired by recent work on foveated rendering where peripheral vision has lower retinal resolution,

rendering cost can be dramatically reduced as well at large eccentricity. However, there is no theoretical guideline on the savings, and prior techniques do not apply to light field sampling. We show that, through Fourier analysis, more theoretical bounds for saving can be revealed in both spatial and angular dimensions.

5.9.2 Analysis of Frequency Bound due to Display

Zwicker et al. [98] have shown that when object extends beyond the depth of field (DoF) of the light field display, the spatial domain is subject to frequency clipping and thus low-pass filtered.

$$B_{\omega_{\mathbf{x}}}^{display} = \begin{cases} \frac{1}{2\Delta\mathbf{u}_d \hat{k}(d_o,f)}, & \text{if } \hat{k}(d_o,f) \geq \frac{\frac{d_e}{d_d}\Delta\mathbf{x}_d}{\Delta\mathbf{u}_d} \\ \\ \frac{d_d}{2d_e\Delta\mathbf{x}_d}, & \text{otherwise,} \end{cases} \tag{47}$$

These bounds are illustrated in Figure 25c.

5.9.3 Sampling Transformation

In Section 5.4.1, each d_ζ from

$$W(\mathbf{x},\mathbf{u}) = \int_{d_\zeta^-}^{d_\zeta^+} w_d(d_\zeta) w_s(d_\zeta) \mathrm{d}d_\zeta, \tag{48}$$

defines an independent coordinate system $(\omega_{\mathbf{x}}, \omega_{\mathbf{u}})$ with the slope $\hat{k}\left(d_\zeta, f_\zeta\right) = 0$. For fast and closed form computation of the integration, we transform them, through operator η, into one uniform coordinate frame such that $\hat{k}\left(d_\zeta^-, f_\zeta^-\right) = 0$ (i.e., relative to the coordinate frame when the eye is focusing at d_ζ^- with focal length f_ζ^-). The transformed d_ζ and $(\mathbf{x},\mathbf{u})$ are defined as d'_ζ and $(\mathbf{x}',\mathbf{u}')$.

In the transformed frequency frame, a point $(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}})$ can be computed as:

$$\begin{aligned} \begin{bmatrix} \omega'_{\mathbf{x}} \\ \omega'_{\mathbf{u}} \end{bmatrix} &= \left(1 + \hat{k}\left(d_\zeta^-, f_\zeta\right)^2\right)^{-\frac{1}{2}} \begin{bmatrix} 1 & \hat{k}(d_\zeta^-, f_\zeta) \\ -\hat{k}(d_\zeta^-, f_\zeta) & 1 \end{bmatrix} \begin{bmatrix} \omega_{\mathbf{x}} \\ \omega_{\mathbf{u}} \end{bmatrix} \\ &\triangleq \eta(d_\zeta, \omega_{\mathbf{x}}, \omega_{\mathbf{u}}). \end{aligned} \tag{49}$$

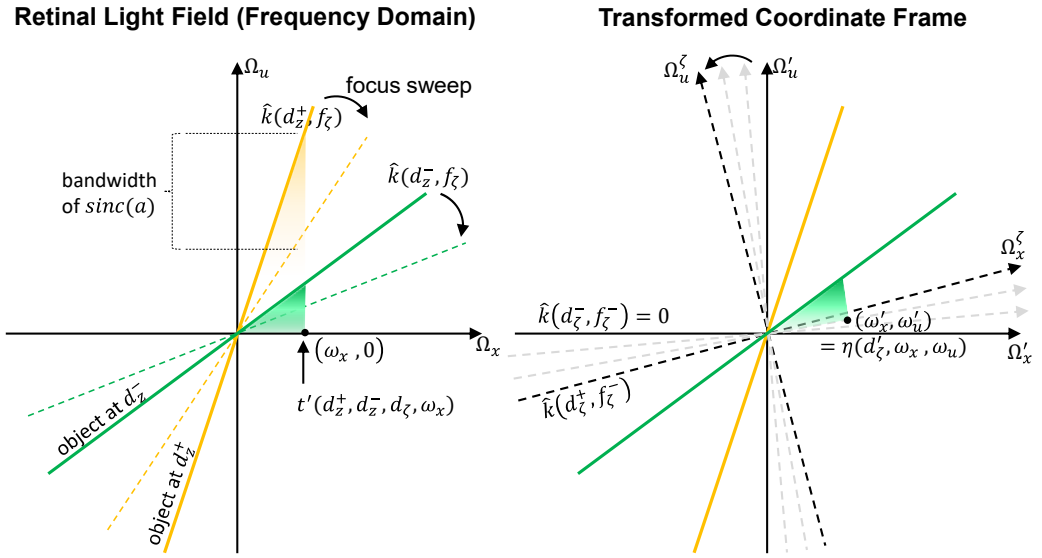


Figure 41: Illustration of importance function and coordinate transformation. *The left figure shows original coordinate system for a given d_ζ before transformation: The (sync-smeared) yellow and green lines represent two object points at different depths $d_z^\pm$. Their perceptual bandwidths $t(d_z^+, d_\zeta, \omega_\mathbf{x})$ and $t(d_z^-, d_\zeta, \omega_\mathbf{x})$ are evaluated at $(\omega_\mathbf{x}, 0)$, and their difference represents $\bar{t}(d_z^+, d_z^-, d_\zeta, \omega_\mathbf{x})$, whose integration (along the Ω_x axis) yields the static weight, $w_s(d_\zeta)$. The dynamic weight $w_d(d_\zeta)$ is similarly integrated but from the rate of change of $\bar{t}$ with respect to d_ζ, i.e. the two lines rotate with varying d_ζ. The right figure shows the transformed system: all coordinates are transformed to the one (Ω_x', Ω_u') respect to d_ζ^-. Correspondingly, all the importance evaluations of d_ζ (transformed as d_ζ') are performed at Ω_x^ζ axis.*

We define its slope as

$$\hat{k} \triangleq \frac{\omega'_\mathbf{u}}{\omega'_\mathbf{x}}. \tag{50}$$

Then its corresponding transformed signal amplitude as

$$\begin{aligned} t'(z_i, \omega'_\mathbf{x}, \omega'_\mathbf{u}) = &\left\| \hat{s}_i \left(-\frac{d_e}{d_{z_i}} \omega'_\mathbf{x} \right) \right\| \\ &\times \operatorname{sinc}\left(a\omega'_\mathbf{x} \left\| \frac{\hat{k} - \hat{k}(d_{z_i}, f_\zeta^-)}{\sqrt{1 + \hat{k}^2(d_{z_i}, f_\zeta^-)}} \right\| \right). \end{aligned} \tag{51}$$

With the formulation above, the static importance defined on the point $(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}})$ is

$$w'_s(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}) = \left\| t'(d_z^+, \omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}) - t'(d_z^-, \omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}) \right\|, \tag{52}$$

and the dynamic importance defined on the line with slope $\hat{k}$ becomes

$$w'_d(\hat{k}) = \int_0^{B_{\mathbf{s}}} \frac{w'_s(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}})}{\partial \hat{k}} \mathrm{d}\omega'_{\mathbf{x}}. \tag{53}$$

Now Equation (48) can be recomputed as:

$$\int \int w'_d(\hat{k}) w'_s\left(\omega'_{\mathbf{x}}, \omega'_{\mathbf{u}}\right) \mathrm{d}\omega'_{\mathbf{x}} \mathrm{d}\omega'_{\mathbf{u}}. \tag{54}$$

This closed form integration is derived in Section 5.9.6.

Note that the display ($B^{display}_{\omega_{\mathbf{x}}}$) and lens ($B^{lens}_{\omega_{\mathbf{x}}}$) spatial bounds may also transform along with η. However, the actual range of $\hat{k}$ under a common light field display is small ($\approx \pm 0.037$ with our prototype), and the major influence in periphery is from the untransformed $B^{retina}_{\omega_{\mathbf{x}}}$, so we keep those two bounds invariant when computing Equation (54).

5.9.4 Minimum Display Sampling

To reach a high perceptual threshold, we allow more rays to be sampled than the minimum number required at locations in the adaptive light field sampling Equation (41). Specifically, we guarantee full sampling in the foveal area (within $5\,\mathrm{deg}$ eccentricity). For periphery, according to our bandwidth guideline, we compute the local budget s_{el} for minimum sampling of the display proportional to the density function of the local retinal bandwidth σ^{-1}(Equation (31)):

$$s_{el}\left(\mathbf{x}_d\right) = s_e \frac{\sigma^{-1}\left(\phi^{-1}\left(\mathbf{x}_d, \mathbf{u}_d\right)\right)}{\int \sigma^{-1}\left(\mathbf{x}\right) \mathrm{d}\mathbf{x}}, \tag{55}$$

where s_e is the total peripheral sampling budget, $(\mathbf{x}_d, \mathbf{u}_d)$ is a ray passing the center of eyebox, and ϕ^{-1} maps the display coordinate to retina space (Equation (44)).

To guarantee perception preservation, we also ensure the number of rays to satisfy the condition where the footprint of a ray (e_b/s_{el}) over the eyebox weighted by the

spatial retinal bandwidth is smaller than the smallest solid angle of the hardware ray $\Delta \mathbf{u}_d$ on the pupil plane:

$$\frac{e_b / s_{el}\left(\mathbf{x}_d\right)}{\sigma(\phi^{-1}(\mathbf{x}_d, \mathbf{u}_d))} \leq \epsilon \frac{\Delta \mathbf{u}_d}{\sigma(\mathbf{x}_{fovea})}, \tag{56}$$

where $\mathbf{x}_{fovea}$ represents the foveal boundary, $\epsilon > 1$ tolerates strong sampling reduction from content-adaptive importance Equation (48). We set $\epsilon = 1.2$ in our experiments.

5.9.5 Occlusion Aware Post-Filtering

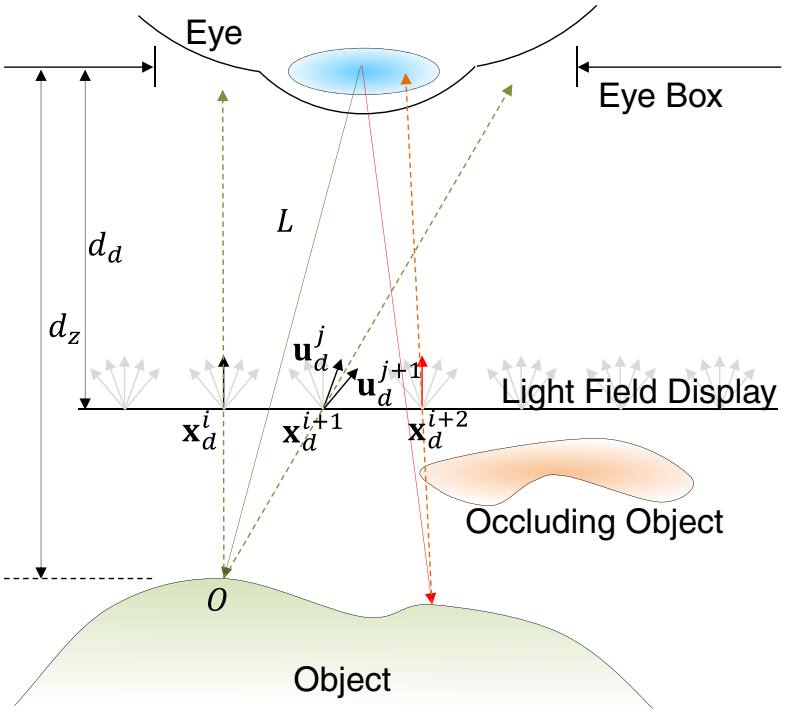


Figure 42: Reconstructing rays for light field display. *The display rays $L_d(\mathbf{x}_d, \mathbf{u}_d)$ can be reconstructed from the sparsely sampled rays L (solid lines) through 4D Gaussain radial basis function by intersecting the reflected rays (dashed lines) to the display pixels.*

The sparsely sampled set of rays is then filtered to be shown on a light field display of rays with uniform spacing. We implement a separable 4D Gaussian radial basis

function for the sparse reconstruction. We first trace the ray $L(\mathbf{x}, \mathbf{u})$ to the scene and intersect it with the point O, and then splat the reflected rays to the the light field display rays $L_d(\mathbf{x}_d^i, \mathbf{u}_d^j)$, as shown in Figure 42, such that their extensions are within the eyebox e_b:

$$\begin{aligned} L_d(\mathbf{x}_d^i, \mathbf{u}_d^j) =& L_d(\mathbf{x}_d^i, \mathbf{u}_d^j) + \mathcal{N}\left(\mathbf{x}_d^i - \phi(\mathbf{x}, \mathbf{u}), \frac{1}{B_{\omega_\mathbf{x}}^{all}(\mathbf{x})}\right) \\ & \times \mathcal{N}\left(\mathbf{u}_d^j - \frac{d_d(\mathbf{x}_d^i - O)}{d_z - d_d}, \frac{1}{B_{\omega_\mathbf{u}}^{all}(\mathbf{x})}\right) \times L(\mathbf{x}, \mathbf{u}) \\ & \forall_{(i,j)} \text{ such that } (\mathbf{x}_d^i + d_d \mathbf{u}_d^j) < \frac{e_b}{2}. \end{aligned} \tag{57}$$

Proper occlusion handling is crucial in the post-filtering that we use the depth map obtained in the first stage of sparse sampling to cull out rays blocked by the occluder, as shown in Figure 42. Finally, similar to Patney et al. [9], a contrast-preserving filter is applied to the rendering.

5.9.6 Closed Form Importance Sampling

To calculate Equation (41) from the transformed frame in Section 5.9.3, we first simplify Equation (37). Because of the small range of $\hat{k}^2(d_{z_i}, f_\zeta^-)$ described in Section 5.9.3, Equation (51) can be approximated and simplified as

$$t'(z_i, \omega'_\mathbf{x}, \omega'_\mathbf{u}) \approx \left\| \hat{s_i}\left(-\frac{d_e}{d_{z_i}}\omega'_\mathbf{x}\right)\right\| \operatorname{sinc}\left(a\omega'_\mathbf{x}\left\|\hat{k} - \hat{k}(d_{z_i}, f_\zeta^-)\right\|\right). \tag{58}$$

Note that we have applied contrast preserving step in the post filtering Section 5.4.2, during sampling stage, we can make an conservative estimation by assuming high frequency amplitude over all surfaces, thus Equation (58) can be further simplified as

$$\begin{aligned} t'(z_i, \omega'_\mathbf{x}, \omega'_\mathbf{u}) &= s_h \operatorname{sinc}\left(a\omega'_\mathbf{x}\left\|\hat{k} - \hat{k}(d_{z_i}, f_\zeta^-)\right\|\right) \\ &\propto \operatorname{sinc}\left(a\omega'_\mathbf{x}\left\|\hat{k} - \hat{k}(d_{z_i}, f_\zeta^-)\right\|\right), \end{aligned} \tag{59}$$

where s_h is a constant amplitude value of high frequency texture. For easier formulation, we define symbols $\hat{k_1} \triangleq \hat{k}(d_{z^-}, f_\zeta^-), \hat{k_2} \triangleq \hat{k}(d_{z^+}, f_\zeta^-)$ for derivations below. Thus $t'(z^-, \omega'_\mathbf{x}, \omega'_\mathbf{u})$ and $t'(z^+, \omega'_\mathbf{x}, \omega'_\mathbf{u})$ can be redefined as $t'(\hat{k}, \hat{k_1}, \omega'_\mathbf{x})$ and $t'(\hat{k}, \hat{k_2}, \omega'_\mathbf{x})$ respectively.

Because of the existence of absolute operator in Equation (38), the integration result relies on relative range of k compared with $\hat{k_1}$ and $\hat{k_2}$ (Intuitive illustration can be seen from Figure 25). That means this is a piece-wise integration. As an example of computation, here we let $\hat{k} \geq \hat{k_2} \geq \hat{k_1}$. Other cases can be derived similarly. Moreover, because of the symmetry of the frequency domain (Figure 25), we can just perform computation for $\omega'_{\mathbf{x}} \geq 0$ w.l.o.g. In this subspace, we have

$$t'\left(\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}}, \hat{k_1}, \omega'_{\mathbf{x}}\right) \geq t'\left(\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}}, \hat{k_2}, \omega'_{\mathbf{x}}\right). \tag{60}$$

The first step is to compute w_d. To equally compare different focus depths, we use same range $\Omega_x = [0, B_{\mathbf{s}}]$. Because values of dynamic weight w_d are small, we estimate their terms through a polynomial approximation of sinc function. Optimal sinc function approximation parameters $\{a_3, a_2, a_1, a_0\}$ have been studied by Qiu et al. [111]:

$$\mathrm{sinc}(x) \approx a_3 x^3 + a_2 x^2 + a_1 x + a_0. \tag{61}$$

Thus we have

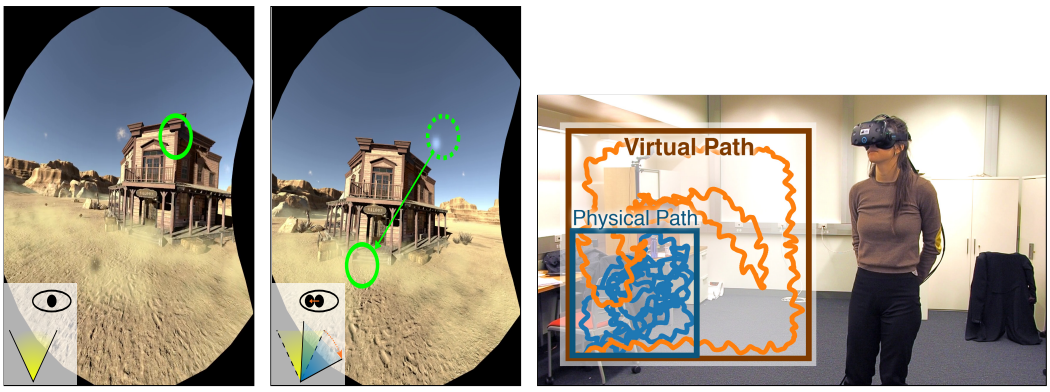
$$\begin{aligned} w'_d(\hat{k}) &\approx \int_0^{B_{\mathbf{s}}} \sum_{i=1}^{2} -1^{i-1} a\omega'_{\mathbf{x}} \left(3a_3(a\omega'_{\mathbf{x}}(\hat{k} - \hat{k_i}))^2 + 2a_2(a\omega'_{\mathbf{x}}(\hat{k} - \hat{k_i})) + a_1\right) \mathrm{d}\omega'_{\mathbf{x}} \\ &\propto \int_0^{B_{\mathbf{s}}} \omega'_{\mathbf{x}} \left(3a_3 a {\omega'_{\mathbf{x}}}^2 (2\hat{k} - \hat{k_1} - \hat{k_2}) + 2a_2\omega'_{\mathbf{x}}\right) \mathrm{d}\omega'_{\mathbf{x}} \\ &\propto 9a_3 a(2\hat{k} - \hat{k_1} - \hat{k_2})B_{\mathbf{s}} + 8a_2. \end{aligned}$$

Using the estimation of w_d above, we obtain

$$
\begin{aligned}
W(\hat{k_1},\hat{k_2}) \propto & \iint \left(9a_3 a\left(2\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}}-\hat{k_1}-\hat{k_2}\right)B_{\mathbf{s}}+8a_2\right) \\
& \left(t\left(\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}},\hat{k_1},\omega'_{\mathbf{x}}\right)-t\left(\frac{\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}},\hat{k_2},\omega'_{\mathbf{x}}\right)\right)\mathrm{d}\omega'_{\mathbf{x}}\mathrm{d}\omega'_{\mathbf{u}} \\
& = \iint \frac{18a_3 a\omega'_{\mathbf{u}}}{\omega'_{\mathbf{x}}}B_{\mathbf{s}}\left(\sum_{i=1}^{2}-1^{i-1}\,\mathrm{sinc}(a\omega'_{\mathbf{u}}-a\hat{k_i}\omega'_{\mathbf{x}})\right)\mathrm{d}\omega'_{\mathbf{x}}\mathrm{d}\omega'_{\mathbf{u}} \\
& + \iint \left(8a_2-9(\hat{k_1}+\hat{k_2})a_3 aB_{\mathbf{s}}\right)\left(\sum_{i=1}^{2}-1^{i-1}\,\mathrm{sinc}(a\omega'_{\mathbf{u}}-a\hat{k_i}\omega'_{\mathbf{x}})\right)\mathrm{d}\omega'_{\mathbf{x}}\mathrm{d}\omega'_{\mathbf{u}} \\
& = 18a_3 B_{\mathbf{s}} \\
& \times \int\left(\sum_{i=1}^{2}-1^{i-1}\left(-\sin(a\omega'_{\mathbf{u}})\,\mathrm{Ci}(a\hat{k_i}\omega'_{\mathbf{x}})+\mathrm{Si}(a\omega'_{\mathbf{u}}-a\hat{k_i}\omega'_{\mathbf{x}})+\cos(a\omega'_{\mathbf{u}})\,\mathrm{Si}(a\hat{k_i}\omega'_{\mathbf{x}})\right)\right)\mathrm{d}\omega'_{\mathbf{u}} \\
& - \left(8a_2-9(\hat{k_1}+\hat{k_2})a_3 aB_{\mathbf{s}}\right)\int\left(\sum_{i=1}^{2}-1^{i-1}\frac{\mathrm{Si}(a\omega'_{\mathbf{u}}-a\hat{k_i}\omega'_{\mathbf{x}})}{a\hat{k_i}}\right)\mathrm{d}\omega'_{\mathbf{u}}
\end{aligned}
\tag{62}
$$

Here Si/Ci is sine/cosine integration function can be approximated through Padé approximant. The integration over $\omega'_{\mathbf{u}}$ can be derived with the help of equation below

$$
\int \mathrm{Si}(a\omega'_{\mathbf{u}}-a\hat{k_i}\omega'_{\mathbf{x}})\mathrm{d}\omega'_{\mathbf{u}}=\frac{1}{a}\left((a\hat{k_i}\omega'_{\mathbf{x}}-a\omega'_{\mathbf{u}})\,\mathrm{Si}(a\hat{k_i}\omega'_{\mathbf{x}}-a\omega'_{\mathbf{u}})+\cos(a\hat{k_i}\omega'_{\mathbf{x}}-a\omega'_{\mathbf{u}})\right)
\tag{63}
$$



(a) *Left eye, before saccade* **(b)** *After saccade* **(c)** *Experimental setup with superimposed top view of physical and virtual paths*

Figure 43: Triggering and harnessing temporary blindness via saccades for room-scale redirect walking in VR. *Our system renders a virtual environment into a pair of HMD views while tracking the user eye gaze. (a) shows a HMD left eye rendering for the viewer with overlaid visualizations of tracked eye gaze (green circle) and view frustum (lower left corner). When saccades (rapid eye movements) and head rotations are detected, our system rotates the virtual environments to redirect the users (b). Such rotations are visible during normal viewing conditions, but can be imperceptible during eye or head movements. (c) photographs our experimental setup with a Vive HMD augmented with SMI gaze tracking. Superimposed are the top view of the recorded movements of the physical path in a 3.5m $\times$ 3.5m real room and the virtual path in a much larger 6.4m $\times$ 6.4m synthetic space. Scene courtesy of NOT_Lonely (Vitaly).*

6 Saccadic Locomotion

Redirected walking techniques can enhance the immersion and visual-vestibular comfort of virtual reality (VR) navigation, but are often limited by the size, shape, and content of the physical environments.

We propose a redirected walking technique that can apply to small physical environments with static or dynamic obstacles. Via a head- and eye-tracking VR headset, our method detects saccadic suppression and redirects the users during the resulting temporary blindness. Our dynamic path planning runs in real-time on a GPU, and thus can avoid static and dynamic obstacles, including walls, fur-

niture, and other VR users sharing the same physical space. To further enhance saccadic redirection, we propose subtle gaze direction methods tailored for VR perception.

We demonstrate that saccades can significantly increase the rotation gains during redirection without introducing visual distortions or simulator sickness. This allows our method to apply to large open virtual spaces and small physical environments for room-scale VR. We evaluate our system via numerical simulations and real user studies.

6.1 Related Work

6.1.1 Redirected Interaction in VR

Redirected interaction, such as walking [6, 38, 40, 112, 113] and touching [114, 115], has received recent attention in the graphics and HCI community as a technique that uses mapping and rendering methods to enhance presence. It works by modifying what the user sees while they are physically interacting with their surroundings [116]. Due to the dominance of vision over other senses, the user perceives the physical interaction as being consistent to the visual stimulus. This way, physical interactions can be redirected. In particular, redirected walking can influence the user's walking path in an imperceptible fashion, simulating larger virtual environments within smaller physical ones and avoiding walls and obstacles. Researchers have proposed two primary methods of redirected walking: those that work by dynamically scaling user motion and head rotation for the virtual camera [14, 38, 39, 116] due to sensory conflicts in virtual environments [19], and those that work by warping the virtual scene [6, 40].

Notwithstanding the specific technique, contemporary redirected techniques assume that users are aware of the environment at all times. The techniques do not consider perceptual masking effects, such as saccades, blinks, and other perceptual suppressions. In this work, we enhance redirected interaction by detecting these masking effects and amplifying redirection during these events without introducing any virtual scene warping.

6.1.2 Gaze-contingent Rendering in VR

Gaze-contingent graphics is a widely studied area with several applications in medicine, optometry, vision science, and computer graphics [117, 118]. However, due to the increasing availability of high-quality eye trackers [119] as well as growing research into potential applications, gaze-contingent rendering has also gained popularity in virtual and augmented reality. When used for foveated rendering, gaze-contingent rendering helps improve visual quality without performance compromises [9, 33, 120–124]. When used to simulate high-dynamic range [125] or to provide focus cues [126], gaze-contingent graphics enable new experiences on contemporary displays. Finally, eye-tracking is a useful interaction tool for virtual environments [127]. Thus, eye-tracking support is foreseen in the next generation commodity VR/AR devices. Our system employs eye-tracking to determine occurrences of perceptual suppression for VR redirected walking.

6.1.3 Saccadic and Blink Suppression

A saccade is the rapid eye movement that occurs when we change fixation points. During normal viewing, saccades occur several times a second, contain extremely fast motion (up to 900 $\mathrm{deg} / \mathrm{sec}$), and are long (20–200 ms) compared to VR frame durations [128], although the detection results may vary according to the chosen algorithms [129]. Saccades are among many behaviors that trigger temporary perceptual suppression. Others include masking by patterns, tactile saccades [130], and blinks [131]. While our system for redirected walking could potentially extend to any of these, we explicitly evaluate it under saccades in this work.

Saccadic suppression (a.k.a. *saccadic omission*) of perception occurs before, during, and after each saccadic eye motion [132]. While the exact mechanism behind it is an area of active research [132–134], the characteristics are well-known [135–137]. Our system exploits the particular documented phenomenon suppression of *image displacement* [138, 139].

A key property of visual saccades is that they are ballistic in nature [128] and their velocity profile and landing position can often be predicted mid-flight [140, 141]. This, in addition to saccadic suppression lasting for a short period after the saccade itself completes, suggests that detecting saccades and altering rendering based on the detection should be fairly tolerant of current VR eye-tracking-to-photon latency of around 35 ms [124]. Recent work established reorientation

and repositioning thresholds for VR during saccades [142] and blinks [143]. We leverage those established perceptual thresholds to build and evaluate a redirected walking system.

6.1.4 Subtle Gaze Direction

Subtle gaze direction (SGD) uses image-space modulation to direct a viewer's gaze to a specific target [144]. When applied in peripheral regions these can direct attention without affecting net perception of the scene. Previous work used SGD to trigger controlled saccades to enhance visual search performance [145, 146] and as a narrative tool [147]. Recent work suggests that SGD can drive user gaze in VR experiences as well [148, 149]. We integrate SGD into our system to dynamically and subtly increase the frequency of saccades, which we then exploit as opportunities for imperceptible transformation of the world.

6.2 Pilot Study of Visual Saccades

The efficacy of redirection during saccadic suppression depends on several factors, including frequency and duration of saccades, perceptual tolerance of image displacement during saccadic suppression, and the eye-tracking-to-display latency of the system. To quantify these, we have conducted a short pilot study with six participants using an HTC Vive HMD with integrated SMI eye-tracking. They were instructed to walk a pre-defined path in the small "Van Gogh room" scene and search for six fixed task objects. We recorded their gaze orientations (Figures 44e and 44f) and used the method of adjustment to identify the angular rotation redirections. Specifically, we tuned the rotation angles up/down until the participants could/could not recognize the difference between saccadic redirection, head-only redirection, and walking without redirection by answering "Yes, I noticed something in the camera orientation" or "No, I do not. They are all normal and the same".

We determined no participant could detect camera rotation of less than $12.6\,\mathrm{deg}\,/\,\mathrm{sec}$ ($0.14\,\mathrm{deg}$ at 90 frames per second) when their gaze velocity was above $180\,\mathrm{deg}\,/\,\mathrm{sec}$. We increase redirection for longer saccades linearly, which is consistent with previous perceptual experiments [139, 142]. Bolte and Lappe [142] have shown that "participants are more sensitive to scene rotations orthogonal to the sac-

cade". However, since our overall system computes across multiple frames (Section 6.4.2), saccade directions may change within this period. To guarantee imperceptibility, we choose a conservative gain threshold assuming orthogonal saccades.

We then augmented the data from our experiment with captured head and gaze orientation recorded from a participant playing commercial VR arcade games *NVIDIA VR Funhouse* (Funhouse) , and horror defense game *The Brookhaven Experiment* (Brookhaven), for 10 minutes each (Figure 44). While less controlled as experimental settings, these represent the state of the art for VR presence, rendering quality, and entertainment tasks. They are more realistic and less-biased to evaluate the potential for redirected walking than our specially-constructed lab scenario. For each frame in the collected data, we used our previously measured gaze thresholds to predict the maximum imperceptible redirection.

Over one-minute intervals, the proportion of redirected frames varied between 2.43% and 22.58% in Funhouse, and between 10.25% and 22.02% in Brookhaven. The average proportion of frames with redirection was approximately 11.40% for Funhouse, and approximately 15.16% for Brookhaven, which can sufficiently provide 1.4 $\deg / \sec$ and 1.9 $\deg / \sec$ angular gains. We conclude that the frequency and distribution of redirection depend on the content, yet contain significant extra gains due to saccadic suppression.

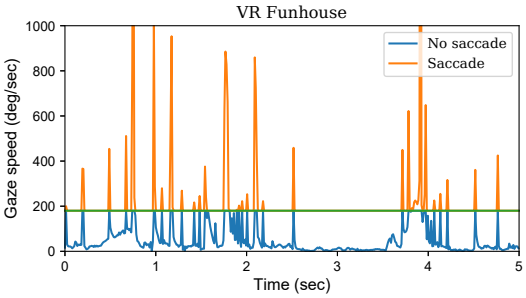
6.3 Method

Reorientation is a technique that modifies the user's virtual camera to decrease the likelihood of exiting the physical play area. Since minor changes in the virtual camera during head rotation are generally imperceptible, this helps provide richer experiences without the user noticing the redirection. Our system also reorients the virtual camera, but it does so not only during head rotations, but also during, and slightly after, eye saccades. Similar to the case with head rotation, small changes to the camera orientation during saccades are imperceptible, and hence offer opportunities for introducing more frequent and greater amounts of redirection.

Our redirected walking method consists of the following three parts:

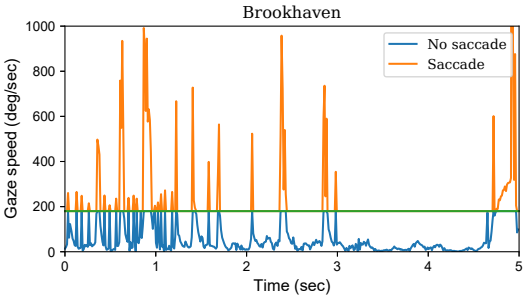
Saccade detection Use gaze tracking to detect saccades and identify opportunities to reorient the virtual camera for redirection (Section 6.3.1).

(a) *NVIDIA VR Funhouse*

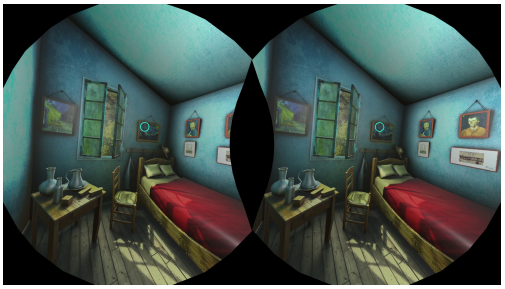


(b) *Gaze plot for (a)*

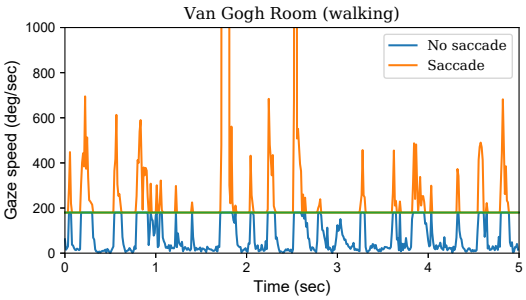
(c) *The Brookhaven*



(d) *Gaze plot for (c)*



(e) *The Van Gogh Room*



(f) *Gaze plot for (e)*

Figure 44: Saccade analysis for VR applications. *We recorded head and gaze data for a user playing two VR games and a simple scene with VR walking to estimate the potential benefits from saccadic redirection. We plot five seconds of angular gaze velocity for these applications, showing frames that we detected as saccades (above the* $180\,\mathrm{deg}\,/\,\mathrm{sec}$ *threshold visualized in green lines). Section 6.1.4 describes our pilot study setup and analysis using (e). Scene (e) courtesy of ruslans3d.*

PathPlanningState = Ready
$\Delta\theta = 0$
function RENDERREDIRECTED(t, $\mathbf{M}(t)$)
    E_{curr} = GETLATESTEYEPOS
    H_{curr} = GETLATESTHEADPOSE
    G_{curr} = COMBINEHEADGAZE(H_{curr}, E_{curr})
    Δ_g = MEASUREANGLE(G_{curr}, G_{prev})
    Δ_h = MEASUREANGLE(H_{curr}, H_{prev})
    Δ_t = GETFRAMEDELTATIME
    if $\Delta_g > 180 \cdot \Delta_t$ **then**
        $\Gamma_g = 12.6 \cdot \Delta_t$
    end if
    if PathPlanningState **is** Ready **then**
        Initialize optimization by sampling $\mathcal{S}$ using Equation (66)
        PathPlanningState = Running
    else if PathPlanningState **is** Running **then**
        Perform iterations of planning optimizer (Equation (71)*)*
        if Optimization is done **then**
            Update redirection angle $\Delta\theta$ (Equation (71)*)*
            PathPlanningState = Ready
        end if
    end if
    if $\Delta\theta > 0$ **then**
        if $(sgn(\Delta\theta)$ **=** $sgn(\Delta_h))$ **then** $\lambda = 0.49$ **else** $\lambda = -0.2$
        $\Gamma_h = \lambda \cdot \Delta_h$
        $\Delta\theta_t = sgn(\Delta\theta) \cdot \min(\|\Gamma_h\| + \|\Gamma_g\|, \|\Delta\theta\|)$
        $\mathbf{M}(t+1) \leftarrow \mathbf{M}(t)$ *and* $\Delta\theta_t$ via Equations (64) and (65)
        $\Delta\theta = \Delta\theta - \Delta\theta_t$
    end if
    if SGDMode **is** ObjectSpace **then**
        Modulate material luminance of selected objects
    end if
    Draw current frame
    if SGDMode **is** ImageSpace **then**
        Modulate luminance of selected peripheral pixels
    end if
    Display rendered frame
    $G_{prev} = G_{curr}$
    $H_{prev} = H_{curr}$
end function

Algorithm 1: Overview of our approach.

Dynamic path planning Use the saccade detection thresholds and the physical space around the user to dynamically determine the best virtual camera orientation for redirection (Sections 6.3.2 and 6.3.3).

Subtle gaze direction (SGD) Render temporally modulating stimuli in a user's visual periphery to induce visual saccades (Section 6.3.4).

Algorithm 1 summarizes the steps that constitute each frame of our approach. During each frame, we first use current and previous gaze orientation to detect visual saccades, identifying the opportunity for redirected walking. We then update our dynamic path planning algorithm, which we amortize over 2–5 frames to maintain real-time performance. After its final iteration, our path planning algorithm returns a direction and magnitude of desired redirection. If the current frame is a candidate for redirection, either due to an ongoing saccade or head rotation, we modify the camera viewpoint in the direction of desired redirection, by a magnitude subject to our perceptual limits. Finally, while rendering the frame, we add subtle gaze direction − temporally pulsating stimuli in a user's peripheral vision to imperceptibly encourage visual saccades. We can apply SGD stimuli in either object space or image space.

6.3.1 Saccade Detection for Camera Reorientation

Saccade detection Once calibrated, our high-speed eye-tracker is relatively noise-free. Thus we use a simple heuristic to determine whether users are currently making visual saccades. At the beginning of each frame, we use the past two gaze samples to estimate the current angular velocity of the user's gaze. If the angular velocity is greater than $180\,\mathrm{deg}\,/\,\mathrm{sec}$, we conclude that a saccade is either currently ongoing or has recently finished.

In our implementation we use the average position of the user's left and right gaze locations. This helps reduce noise in detecting location and in estimating velocity. More robust detection (e.g., *Hidded Markov Model* or *Hidded Markov Model* [129]) are potential future research for lower-quality tracking devices.

Due to the latency of contemporary eye-trackers as well as VR rendering and display pipelines, saccade detection generally lags actual saccades by tens of milliseconds. However, since the duration of visual saccades ranges from 20–200 ms and saccadic suppression lasts for 100 ms after a saccade begins [136, 137], we

find that our detection is relatively tolerant of tracking and rendering latency, especially for saccades with large angular amplitude. Our pilot studies as described in Section 6.1.4 indicated that the empirically-determined threshold of $180\,\mathrm{deg}\,/\,\mathrm{sec}$ accounts for this tolerance.

Camera reorientation thresholds When saccades are detected within a frame, we slightly re-orient the virtual camera by up to $0.14\,\mathrm{deg}\,/$ frame as described in Section 6.1.4. If we respect this threshold, our path planning algorithm can successfully perform redirections with meaningful direction and magnitude without alerting the user. Saccadic redirection can be combined with conventional head-only reorientation. For the latter, we use the previously studied angular gain threshold within $[-20\%, 49\%]$ [39] pre-calibrated within this range for individual users as some may have lower detection thresholds than others [150]. Although rotation during head movement allows more redirection, large head rotations are less frequent than large saccades, so we expect an overall improvement by using both for redirected walking. The saccadic detection threshold $180\,\mathrm{deg}\,/\,\mathrm{sec}$ and gain speed $12.6\,\mathrm{deg}\,/\,\mathrm{sec}$ were set through our pilot study (Section 6.1.4).

6.3.2 Dynamic Path Planning

The saccade-guided camera manipulation and subtle gaze direction (SGD) facilitate VR redirected walking. However, to guide users away from both stationary and moving obstacles, the system must dynamically compute the virtual camera orientation in each frame. Existing off-line mapping approaches [6, 40] require slow pre-processing, incompatible with saccadic actions that happen dynamically and unpredictably in real time. We would also like to avoid any visual distortion caused by virtual scene warping and rely only on larger, rigid transformation gains enabled by saccadic suppression. Thus, we present a real-time dynamic path planning approach driven by perceptual factors (such as SGD), scene properties (e.g. floor layouts and scene object placements), and GPU parallelization.

Formulation For a given frame t and a 2D virtual position $\mathbf{x} = (x, y)$, we model the corresponding physical position $\mathbf{u} = (u, v)$ using an affine transformation $\mathbf{M}$

between the virtual and physical spaces:

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \mathbf{M}(t)\left(\mathbf{x} - \mathbf{x}_c(t)\right) + \mathbf{x}_c(t) \\ \mathbf{M} &= [\mathbf{R}|\mathbf{T}] \end{aligned} \tag{64}$$

where $\mathbf{x}_c(t)$ is the user's current virtual space position. This formulation interprets $\mathbf{x}$ and $\mathbf{u}$ as the next virtual and real user positions to allow optimization for the near future, such as avoiding obstacles.

The goal of the real-time path planner is to find the next frame's optimal translation $\mathbf{T}(t+1)$ and rotation $\mathbf{R}(t+1)$ components so that the redirected walking path during saccades can guide users away from boundaries and obstacles. In our initial investigations we have found $\mathbf{R}$ to be much more effective than $\mathbf{T}$ with saccades and head rotations, so we set $\mathbf{T}(t) = 0$ to reduce the real-time, multidimensional computation workload:

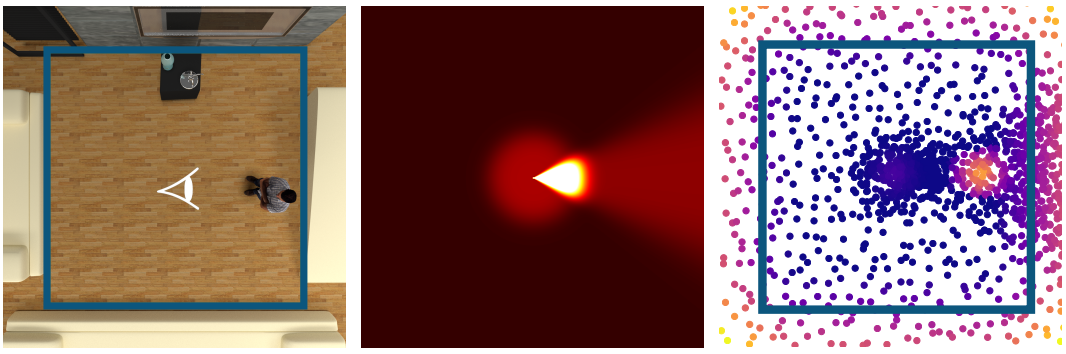
$$\mathbf{M}(t+1) \leftarrow \begin{bmatrix} \cos\left(\Delta\theta(t)\right) & -\sin\left(\Delta\theta(t)\right) \\ \sin\left(\Delta\theta(t)\right) & \cos\left(\Delta\theta(t)\right) \end{bmatrix} \mathbf{M}(t) \tag{65}$$

where $\Delta\theta$ is the redirection angle to optimize for (Section 6.3.3).

Dynamic sampling Inspired by [6, 40], we perform optimization via virtual scene samples. However, instead of global uniform sampling, we dynamically allocate the sample set $\mathcal{S}$ locally, adapting to the user's position and orientation to enhance optimization quality and speed. Specifically, we design an importance-based real-time sampling mechanism emphasizing areas that are (1) close to the user's current position and (2) visible and within the user's current camera frustum, to predict possibilities in the nearer future, as exemplified in Figure 45. To achieve fast results, we created a closed-form formulation for the intuition above. The importance is computed in the polar coordinates $(r(\mathbf{x}), \theta(\mathbf{x}))$ of the virtual space with $\mathbf{x}$ as the origin:

$$\begin{aligned} I(\mathbf{x}) = &\left(-\operatorname{erf}\left(\alpha_0^r r(\mathbf{x}) + \alpha_1^r\right) + \alpha_2^r\right) \\ &\times \left(\exp\left(-\frac{\left(\cos(\theta(\mathbf{x}) - \theta_c) - 1\right)^2}{\alpha_0^a}\right) + \alpha_1^a\right) + \alpha^o \end{aligned} \tag{66}$$

where $\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt$ is the error function (https://en.wikipedia.org/wiki/Error_function), θ_c is the user's current virtual camera direction, $\alpha^r_{i\in\{0,1,2\}}$ and $\alpha^a_{i\in\{0,1\}}$ are parameters fitting to a given space size, and α^o is added to avoid zero samples at low



(a) *physical room* **(b)** *importance heat map* **(c)** *samples*

Figure 45: Illustration of Equations (66) and (71). *Suppose the user is standing at the center of the room while facing right, as shown in (a). The room contains a static obstacle (table on the top) and a dynamic obstacle (person on the right). (b) shows the sampling importance heatmap from Equation* (66)*. (c) plots the corresponding samples. Their color gradients represent energy values from Equation* (71)*. Energies are high for samples outside the physical space and close to the moving human obstacle. Energies are low around the table, because it is far from the user's current position and orientation. The dark blue rectangle in (a) and (c) shows the available physical space.*

importance areas. The importance value is higher at areas close to the user's current position (smaller r) and orientation (θ closer to θ_c,). This is illustrated in the heat map and the corresponding sample set $\mathcal{S}$ in Figure 45. To obtain uniform sampling parameters, we numerically normalize the virtual space to a 1×1 unit. In this space, we use $\alpha_0^r = 30, \alpha_1^r = -3, \alpha_2^r = 1.15, \alpha_0^a = 0.01, \alpha_1^a = 0.1, \alpha^o = 0.01$ in our experiments. Implementation details of performing the sampling are described in Section 6.4.2.

Based on $\mathcal{S}$, we propose the following energy terms that guide users away from physical boundaries and obstacles, keep the redirection from being noticeable by the users, and respond to dynamic user saccades and environment changes in real time.

Static boundary avoidance Similar to [6, 40], the redirection should automatically help users avoid static physical boundaries like walls. We adapt the soft

barrier function from [40]:

$$E_B(t, \Delta\theta) = \sum_{\mathbf{l}_i} \sum_{\mathbf{x}(t) \in \mathcal{S}} w_b\left(\mathbf{x}(t)\right) \left(d(\mathbf{u}, \mathbf{l}_i) + \sqrt{d(\mathbf{u}, \mathbf{l}_i)^2 + \epsilon} \right)^{-1} \\ \mathbf{u} \triangleq \mathbf{u}(\mathbf{x}(t), t+1) \tag{67}$$

where $\mathbf{l}_i$ is the i-th edge of the physical boundary polygon, $d(\mathbf{u}, \mathbf{l})$ is the distance between user's real position $\mathbf{u}$ and boundary edge $\mathbf{l}$, and $\mathbf{u}(\mathbf{x}, t+1)$ is a function of $\Delta\theta$ (Equations (64) and (65)). The term $w_b(\mathbf{x})$ weighs $\mathbf{x}$'s importance for boundary avoidance. Intuitively, w_b should emphasize the virtual samples closer to current user's virtual position $\mathbf{x}_c$, since the user will more likely reach those points. We fit w_b as an exponential function of the distance $d(\mathbf{x}, \mathbf{x}_c)$ between $\mathbf{x}$ and $\mathbf{x}_c$:

$$w_b(\mathbf{x}) = \exp(-d(\mathbf{x}, \mathbf{x}_c)^2 / \alpha_0^b) + \alpha_1^b, \tag{68}$$

where α_0^b is used to ensure that the weights $w_b(\mathbf{x})$ are appropriate for the size of the virtual space and α_1^b is used to avoid zero weights. We use $\alpha_0^b = 0.01, \alpha_1^b = 0.002$ in our experiments. We further calculated w_b from $\mathcal{S}$ which prioritizizes virtual regions that are closer to the current user position and orientation (Equation (66)). Note that Equation (67) represents physical boundaries as polygon clusters and thus can handle non-convex or curved shapes via polygonization.

Moving obstacle avoidance One major limitation of previous redirected walking approaches is the inability to handle dynamically moving obstacles like other people in the same physical room [116]. Our dynamic sampling and GPU accelerated redirection planning let our redirection respond to such real-time physical environment changes.

To analytically model obstacles and obtain high gradients at barrier edges, we use a weighted error function instead of the Gaussian barrier function in [6] to guide users away from obstacles:

$$E_O(t, \Delta\theta) = \sum_{\mathbf{o} \in \mathbf{O}} \sum_{\mathbf{x} \in \mathcal{S}} w_o(\mathbf{x}, \mathbf{u}^{\mathbf{o}}) \operatorname{erf}\left(\alpha_0^m(r^{\mathbf{o}}) \left\| \mathbf{u}(\mathbf{x}, t) - \mathbf{u}^{\mathbf{o}} \right\|^2 + \alpha_1^m \right) \tag{69}$$

where $\mathbf{O}$ is the set of obstacles, $\{\mathbf{u}^{\mathbf{o}}\}$ and $\{r^{\mathbf{o}}\}$ are the dynamic position and radius of each obstacle $\mathbf{o}$, and the linear parameters α_0^m and α_1^m are used to fit the sizes of the obstacles with regard to the erf function. We set $\alpha_0^m < 0$ so that E_O is lower

for $\mathbf{u}(\mathbf{x}, t)$ further away from $\mathbf{u^o}$. The obstacle avoidance parameters $\alpha_{0,1}^m$ should adapt to the obstacle sizes to properly guide users away from potential collision. Specifically, we let $\alpha_0^m(r^\mathbf{o}) = \frac{-1}{r^\mathbf{o}}, \alpha_1^m = 2$. Since dynamic obstacles tend to be smaller than wall boundaries, for efficiency and to reduce potential interference with Equation (67), we consider the obstacles only when users are nearby:

$$w_o(\mathbf{x}, \mathbf{u^o}) = \begin{cases} 1 - \frac{d(\mathbf{x}, \mathbf{x}_c)}{2r^\mathbf{o}} & d(\mathbf{u}, \mathbf{u^o}) < 2r^\mathbf{o}, \|\theta(\mathbf{x}) - \theta_c\| < 15\deg \\ 0 & otherwise \end{cases} \tag{70}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t, \Delta\theta)$ is the redirected physical position of $\mathbf{x}$ at the current time t.

6.3.3 Real-time Optimization and Redirection

Given the energy terms above and a given time frame t, the optimal redirected mapping is defined as

$$\arg\min_{\Delta\theta} E(t, \Delta\theta) = E_B(t, \Delta\theta) + wE_O(t, \Delta\theta). \tag{71}$$

We set $w = 500$ in our experiments. The visualization of the object among each sample in $\mathcal{S}$ can also be seen from Figure 45c.

Dynamic path planning Our system applies only rigid rotation from the optimized $\Delta\theta(t)$ during saccades and head rotations. Not having a distortion energy term makes it simpler to optimize than warping-based methods [6, 40].

Note that the perceptually unnoticeable angular gain from saccade suppression is limited to $[-\Delta\theta_{\max}, \Delta\theta_{\max}]$, where $\Delta\theta_{\max}$ is $12.6\deg/\sec$ in Section 6.1.4. To match this constraint while obtaining real-time performance responding to users' dynamic saccadic actions, we implement the optimization as a GPU-based line searching method, details and performance comparison are shown in Section 6.4.2 and table 4. It is based on the iterative cubic + quadratic zoom searching method with Wolfe condition [151]. With the optimized $\Delta\theta$, we redirect the virtual camera when saccades and/or head rotations are detected.

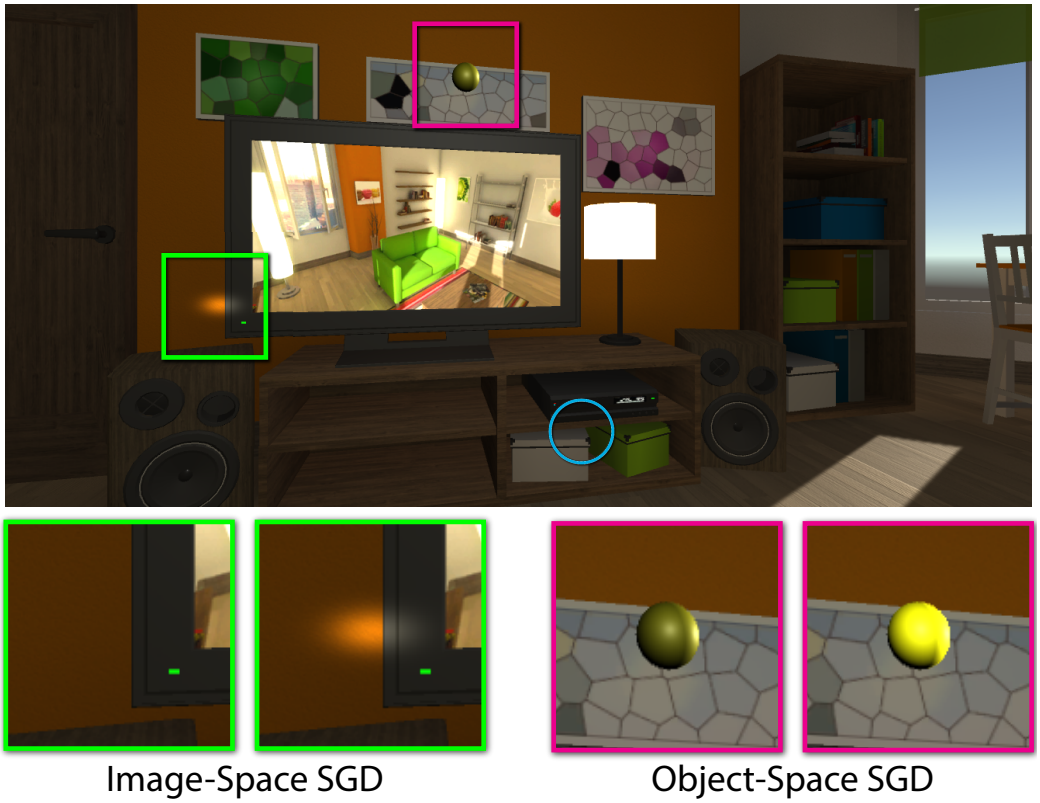


Figure 46: Subtle Gaze Direction (SGD) stimuli used in our study. *This example illustrates the stimuli used in our implementation of subtle-gaze direction. Green inset shows an example of image-space SGD stimulus, and magenta inset shows an example of object-space SGD stimulus. Blue circle indicates the user gaze. Scene courtesy of Barking Dog.*

6.3.4 Subtle Gaze Direction for Saccades

The frequency and extent of visual saccades vary with user, content, and task. However, they directly influence the opportunity for saccadic suppression. Thus, in order to improve the effectiveness of saccadic redirected walking, we would like to increase the occurance of saccades. In the spirit of using distractors to improve redirected walking [152, 153] without introducing noticeable content change, we propose to utilize subtle gaze direction (SGD) [144] to encourage saccades.

Instead of guiding users to look at particular objects or regions, as is the goal of conventional SGD, our primary goal is to encourage larger and more frequent saccades. Hence, we place SGD stimuli as temporally-varying luminance modulations at a user's peripheral vision, as inspired by Grogorick et al. [148]. The radius of our stimulus is $3.5\,\mathrm{deg}$ with a smooth Gaussian fall-off.

Following Sridharan and Bailey [154], we prioritize SGD target locations to overlay objects and image features that are already visually salient. We can select these locations in two different ways, which we call *image-space SGD* and *object-space SGD*.

Image-space SGD finds salient peripheral pixels in the rendered image of the current frame, as shown in Figure 46. Using visual contrast as the saliency measure, we implement image-space SGD by selecting regions with high local contrast to ensure GPU efficiency. To further speed up the search, we down-sample the image via MIPMAP. Section 6.4.1 describes details of our implementation. Our preliminary studies suggested that image-space SGD stimuli in a walking experience are either too hard to perceive or too prominent and hence undesirably distracting. We believe this is because existing SGD mechanisms for either stationary desktop [144] or relatively static AR [155] and VR [148] scenarios may not suffice for highly dynamic redirected walking.

Thus, we also implement object-space SGD, a method that performs SGD modulation directly on the textures/materials of chosen virtual objects, so the users will perceive them as actual scene motion/appearance modulations instead of rendering artifacts. Our object-space SGD approach is straightforward. For each frame, we find scene objects that belong to manually chosen set (e.g. targets of our task), and modulate the color of their diffuse material. To ensure subtlety of SGD, we can choose to only apply SGD to objects that lie in a user's peripheral vision, or to those that are close to the user's current virtual position. Sampled stimuli are

shown in Figures 46 and 46. Note that in our pipeline, object-space SGD works by modifying materials *before* we begin drawing a frame, while image-space SGD works by modifying pixels *after* drawing a frame. Since image-space and object-space SGD approaches are orthogonal, they can be combined for evaluation.

6.4 Implementation

Our system is implemented using an eye-tracked HMD – an HTC Vive augmented with an SMI eye tracker with 250Hz update and 6.5ms response latency, driven by a desktop computer with one NVIDIA Titan Xp GPU, an Intel i7-7700K CPU, and 32GB RAM. For implementing our redirected walking methods in a real-time VR rendering environment, we used the Unity Pro engine, the redirected walking toolkit [114], ShaderLab, and DirectX HLSL pixel and compute shaders.

6.4.1 Subtle Gaze Direction

Image-space SGD Our image-space SGD approach involves applying temporal modulations to pixels in a user's visual periphery. To improve its effectiveness, we use a content-aware approach that prioritizes high-contrast image regions for stimulus placement. Searching for pixels with high local contrast can be an expensive per-frame computation. For acceleration, we compute contrast on a downsampled version of the current frame, which we obtain by generating MIPMAPs for the current frame. After estimating and finding the region with maximum local contrast, we generate the SGD stimulus by modulating the luminance of a Gaussian-shaped region around the center of the high-contrast region. Algorithm 2 provides an overview of this approach.

Object-space SGD We perform object-space SGD as luminance modulations on the diffuse textures/materials of specific scene objects. In general, we would like to select a salient object as the target of SGD. For our study, we simply chose SGD objects from the set of target objects used in our search task, while restricting the set to only those objects that are close to the user's virtual viewpoint.

1: I: current frame (rendered, but not displayed)
2: **function** IMAGESPACESGD(I)
3: Compute MIPMAPs for I
4: Select the 5th MIPMAP image I_5
5: Compute the local Weber contrast for each 3×3 tile in I_5
6: Find *peripheral* pixel $p_5^{max} \in I_5$ with max local contrast
7: Locate the tile t_{max} in I corresponding to p_5^{max}
8: Perform SGD modulation centered at t_{max}
9: **end function**

Algorithm 2: Image-space SGD. *Our approach image-space SGD approach searches for high-contrast regions in a down-sampled version of the currently rendered frame. We use the center of the tile with highest contrast as the center of our stimulus.*

6.4.2 GPU-Based Sampling and Line Search

Performing summation for importance-based samplings over all virtual space areas is slow on the CPU. For fast parallel processing, we distribute the importance sampling task in each local virtual space area into threads in the GPU, each of which performs sampling independently and adds the importance values atomically. Then the overall sampling budget, which depends on GPU capability, is distributed to each thread based on their local value. The portion is computed by dividing the sum of all areas. In our experiments, the budget was set as 500. This significantly reduces the sampling time, to less than $5ms$. This step takes only 1 frame.

In the line searching step, we adapt a searching approach with strong Wolfe condition [151] to find the optimal redirection angle $\Delta\theta$ by minimizing Equation (71). Since the computation of objective Equation (71) and its derivatives of each sample in $\mathcal{S}$ are independent of each other, we also parallelize the computation of each virtual space sample as a thread in the GPU with atomic operation. The parallelization reduces the computation time to $< 5ms$ per iteration. However, line search is an iterative process, which multiplies the computation time of the objective and derivative calculation. To leverage the high VR rendering refresh rate ($90FPS$ for HTC Vive), we distribute the iterations into multiple consecutive frames. In the final system, we perform 2 iterations per frame. This amortizes the path planning optimization over 2–5 frames to maintain real-time performance.

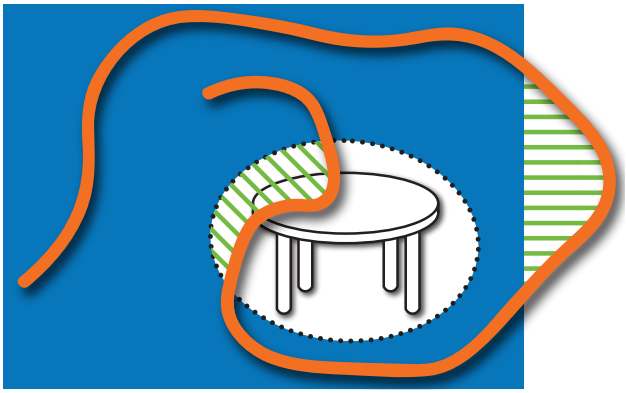


Figure 47: Error measure for user studies. *We compute the error ϵ in Equation* (72) *as the total area (shown striped) that is out of bounds or within obstacles.*

6.5 Evaluation

We evaluated our method with two user studies (Sections 6.5.2 and 6.5.3) and several simulations (Section 6.5.4). The study participants were randomly chosen from internal and external volunteers. One of them was aware of the research, but not the study hypothesis. The study was conducted in a much larger physical space, with a subset of that space designated as the bounds for our redirected walking method. This ensured participant safety without worst-case stimuli, provided a continuous experience, simulated a challenging small room, and facilitated error measurement whenever a participant strayed outside the bounds.

The results show that our method provides significant improvements to redirected walking in VR. We also examine the impact of the three key aspects of our method, saccadic redirection, dynamic path planning, and the use of SGD. While our user studies help to understand the practical effectiveness of our method and identify possible VR sickness, our simulations evaluate our method across a much broader set of conditions with a controlled, consistent set of synthetic inputs and recorded walk-throughs.

6.5.1 Measurement

In the study, we record participants' real/virtual-world positions as well as head orientations and gaze positions. We then visualize the virtual and physical path of each trial, and compute the *error area* for each path—the area outside the physical space or inside an obstacle, as shown in Figure 47. The measure combines the effect of path length with how far each position is from the boundary. With equal

path length, a redirected walking technique is more effective by bringing users back after shorter excursions than guiding them far away from boundaries.

To quantify the effectiveness of the redirection, we compare the error area for the virtual path (without redirection) to the area for the physical path. Smaller ratios indicate more effective redirection. Specifically, we define the effectiveness of the redirected walk as the saving ratio ξ, defined as:

$$\epsilon(p(t \rightarrow \mathbf{u})) = \int \overset{\oplus}{\min_{\mathbf{l}_i}} d_s\left(\mathbf{l}_i, \mathbf{u}\left(t\right)\right) \mathrm{d}t \tag{72}$$

$$\xi = \frac{1 - \epsilon(p_r)/\epsilon(p_v)}{\int h\left(t\right) \mathrm{d}t} \tag{73}$$

, where p is a given physical path that maps a given time t to a physical position $\mathbf{u}(t)$; p_v and p_r are the paths without and with redirection respectively, as visualized in Figure 43c. $\overset{\oplus}{\min}$ finds the minimum non-negative signed distance d_s (positive/negative for outside/inside the real space domain) between exterior-or-interior boundary segment $\mathbf{l}$ and real user position $\mathbf{u}$, and $h(t)$ is the user's head rotation angle at time frame t. ϵ is the total area that is out of bounds or within obstacles. The saving ratio ξ shows how much a redirected path can reduce the error cost compared with the original virtual path overlaid on the real environment. Since we used the savings from head-only gain as the baseline, we normalized ξ by the total head rotations, as users may have a different number of head rotations for multiple trials with different virtual paths.

6.5.2 User Study: Impact of Saccades

Overview In our first user study, we evaluate whether the use of saccades with and without traditional image-space SGD [148] can improve the effectiveness of a redirected walking system. We instructed participants to perform a typical target-retrieval task. Each participant's goal was to search and count all instances of a specific target object in a VR environment.

Task and Stimuli The study consisted of three experiments, all using our dynamic path planning algorithm as the redirection method. Each user did each experiment once.

1. Non-saccadic redirected walking, with head rotation gain only (NON-SACCADE);

2. Saccadic redirected walking (SACCADE);

3. Saccadic redirected walking with image-space SGD from Sections 6.3.4 and 6.4.1 (IMAGE-SGD-I).

Figure 46 shows a screenshot of the VR environment and task stimuli used in this study. Each participant started from the same corner in the virtual room and was instructed to find as many instances of randomly colored, positioned, and scaled balls as possible. Each trial lasted 80 seconds. The sizes of the virtual and physical room were 3.1m×3.1m and 2.0m×2.0m respectively. To encourage walking during retrieval, we dynamically control the transparency of target objects based on their distances to the current user position $\mathbf{x}_c$. Specifically, for the i-th target at position $\mathbf{x}_t$, we update its material alpha (a_i) at time t as

$$a_i(t) = \exp(-\left\|\mathbf{x}_c(t) - \mathbf{x}_t\right\|^2 / 0.05). \tag{74}$$

Prior studies have used fog for a similar purpose [113]. While it is a good alternative, we opt for object transparency so that the overall environment is consistently visible at all times.

At the end of each trial, each participant was asked to complete the Kennedy Lane SSQ [64] for simulator sickness. After the 3 trials, the participant was asked, "Did you notice any camera modulation or difference among all trials?".

Participants 9 users (3 female, 33.3%) participated in the study. The average age was 26.7 ($SD = 1.66$). The median of self-reported experiences with VR was 4, with 1 being least familiar, and 5 being most familiar. We adopted a within-subject design. The order of the three experiments were counterbalanced across participants. Subjects were not informed of the study hypothesis. Between successive trials, a mandatory 3-minute break was enforced.

Results We statistically analyze the recorded error measures among the SACCADE, NON-SACCADE and IMAGE-SGD-I experiments from the study.

Saving ratio The introduction of extra rotation during saccade enables more opportunities to perform stronger angular manipulation, thus smaller physical space usage. To numerically evaluate the capability we compare NON-SACCADE to SACCADE. The average saving ratio ξ was $2.01e-3$ ($SD = 1.95e-3$) for NON-SACCADE, and $3.39e-3$ ($SD = 1.98e-3$) for

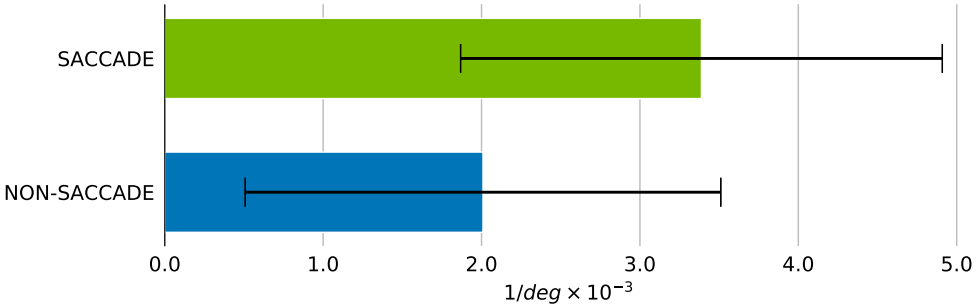


Figure 48: The effects of saccades for redirected walking. *We plot the average saving error ratios ξ (Equation* (72)*) and* 95% *confidence intervals over all users in Section 6.5.2. Notice that saccades can provide stronger error saving for redirected walking than head-rotation only. The confidence error bar indicates that the exact gains (thus variance) likely vary across users and within experiences.*

SACCADE, as shown in Figure 48. There was a significant main effect of SACCADE on ξ ($F_{1,8} = 15.01, p < 0.005$).

Saccadic angular gains To evaluate the impact of SGD, we calculated the sum of all saccadic angular gains between SACCADE and IMAGE-SGD-I. The total saccadic redirected angle across all users was $163.82\,\text{deg}$ ($SD = 28.79\,\text{deg}$) for SACCADE, and $148.63\,\text{deg}$ ($SD = 22.99\,\text{deg}$) for IMAGE-SGD-I. Single factor repeated measures ANOVA did not show a significant main effect of SGD on the saccadic angular gains ($F_{1,8} = 3.306, p = 0.107$).

Subjective feedback Scores below 2 were reported by all users to be caused by nausea and oculomotor, except for one user who reported to have often experienced VR perceptual anomalies including general discomfort, nausea, and vertigo, as shown in the left half of Figure 49. All users answered "no" to the post-trial question, indicating that the saccadic redirection was perceptually unnoticeable.

Discussion The ξ between SACCADE and NON-SACCADE indicates that SACCADE can greatly help redirected walking by reducing errors by 68.7% on average. It is better in performance than NON-SACCADE and in comfort than a latest redirected walking method (Figure 49).

Figure 50 compares redirection methods, with 50a and 50d demonstrating that

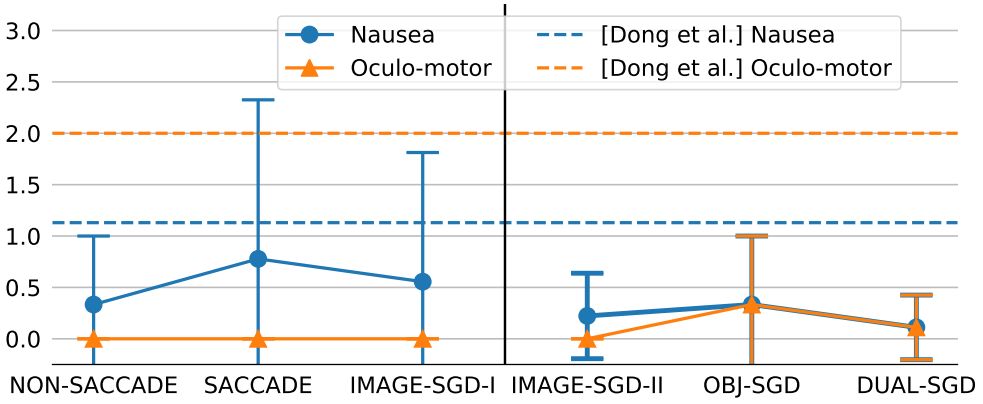


Figure 49: Simulator sickness from Section 6.5.2 and Section 6.5.3. *The dashed lines show the minimum sickness rates from [40]. Our system is shown to reduce sickness comparing to warping-based approach.*

saccadic redirection reduces the chance of hitting physical boundaries, allowing much larger differences between virtual and physical environments.

Image-space gaze direction cues did not trigger saccades for all study subjects in the search tasks. From the saccadic angular gains results we can conclude that while gaze direction cues in general can help trigger saccades in desktop and VR displays (based on our initial development with sitting/static setups), previously reported image-space methods [144, 148] may not be as effective in the highly dynamic redirected walking scenario, especially when it involves search and retrieval tasks (e.g., in real VR games). This observation was also derived from our post-interview with users: most reported that they were focusing on the task object retrieval while constantly moving, thus paying much lower attention to or ignoring the detailed image content, which changes rapidly but contains the image-space SGD stimuli. The result and discovery inspired us to explore an task-matching, object-space SGD variant that we used for a follow-up user study.

6.5.3 User Study: Image-space SGD Vs. Object-space SGD

Overview As described above, for highly dynamic redirected walking applications, image-space SGD stimuli were often not as effective as they are in relatively static scenarios like image viewing [144] or searching while being seated

[148]. We conducted a second study with object-space SGD as described in Section 6.3.4. Using a similar setup as in Section 6.5.2, we evaluated the relative effectiveness of image-space and object-space SGD in increasing the frequency of saccades.

For image-space SGD, we applied SGD stimuli using the algorithm in Section 6.4.1; for object-space SGD we simply modulated the target objects' luminance. The study consisted of three experiments:

1. Image-space SGD only (IMAGE-SGD-II, Figure 46);
2. Object-space SGD only (OBJ-SGD, Figures 46 and 46);
3. Both object-space and image-space SGD (DUAL-SGD).

Participants Another 9 users (2 female, 22.2%) participated in the study. The average age was 26.7 ($SD = 2.24$). The median of self-reported experiences with VR was 3, with 1 being least familiar, and 5 being most familiar. The order of the three experiments were counterbalanced across participants. A mandatory 3-minute break was enforced between successive trials.

Results We compared the effect from different SGD approaches.

Saccadic angular gain The total saccadic angle gain across all users was $145.13\,\mathrm{deg}$ ($SD = 21.83\,\mathrm{deg}$) for IMAGE-SGD-II, $156.78\,\mathrm{deg}$ ($SD = 23.41\,\mathrm{deg}$) for OBJ-SGD, and $167.48\,\mathrm{deg}$ ($SD = 22.56\,\mathrm{deg}$) for DUAL-SGD. There was a significant main effect of SGD method on the total redirected angles ($F_{2,16} = 6.417, p < 0.05$). Pairwise comparison with Holm correction showed the differences between DUAL-SGD and the other two experiments were significant ($p < 0.05$ for both experiments), but not between IMAGE-SGD-II and OBJ-SGD ($p = 0.168$).

Subjective feedback No users noticed any camera modulation. All users reported nausea below 2 and oculomotor below 3, as shown in the right half of Figure 49.

Discussion Compared with traditional image space SGD, the object-plus-image space SGD achieved better results. This shows that in a highly dynamic redirected

walking VR scenario, the impact of image-space SGD becomes weaker. However, having the task objects with similar flickering appearances to image SGD might trigger more saccades since users were looking for such stimuli. Task-dependent SGD design can be an interesting direction for future, more exhaustive studies.

The user perception of saccadic redirection was similar to the first study. Saccadic redirection using the parameters we selected in Section 6.1.4 was imperceptible in our VR exploration and object-retrieval task. Further, since we used head-pose redirection from Steinicke et al. [39] in conjunction with saccadic redirection, we can infer no perceptual impact of the two working together.

6.5.4 Simulation: Redirection Methods

In addition to user studies, we conducted simulations to evaluate our method over a wider set of conditions but using a consistent set of input virtual paths and head orientations for fair comparison. During each study trial, we recorded virtual user position $\mathbf{x}$, head rotation angles, and gaze point of regard in each time frame t. We use recorded rather than procedurally generated user paths for better realism. Although saccadic redirection was enabled while recording, for simulation we used only users' virtual paths which are solely dependent on object placement and the individuals' virtual movements, to avoid bias toward or against any particular redirection approach, such as Steer-to-Center (S2C). The path planners then return the corresponding $\Delta\theta$ values (0 for methods not considering eye/head rotation), allowing us to update $\mathbf{M}(t+1)$ and to get the simulated physical position $\mathbf{u}(t+1)$ at the recorded $\mathbf{x}(t+1)$ using Equation (64). With this mechanism, we can simulate different physical paths with different path planners and/or angular gains, based on the same virtual path as another trial. Error measure analysis (Equation (73)) can also be performed on the new physical path. When simulating virtual spaces with difference sizes, by assuming the same walking speeds, we can rescale the recorded virtual coordinates and insert extra time frames by interpolation.

Dynamic path planning versus S2C Measuring path planning approaches is sensitive to a specific user's virtual traveling path for each trial. To obtain a fair comparison, we simulate S2C redirection results with a same user movement history, as described in Section 6.5.4.

With all 18 users from Sections 6.5.2 and 6.5.3, the average ξ was $3.06e{-}3$ ($SD = 1.52e{-}3$) for the dynamic path planning condition, and $0.75e{-}3$ ($SD = 2.12e{-}3$) for the corresponding simulated S2C condition, as shown in Figure 51. Saccadic suppression had a significant main effect on ξ ($F_{1,17} = 26.12, p < 0.005$). Because eye actions such as saccades occur frequently and uniformly, as shown in Figure 44, it allows stronger and more uniform rotation gains. Figure 50a and Figure 50e compare results with the same user history.

Obstacle and multi-user collision avoidance Traditional redirection planning approaches such as S2C [113, 156] handle convex-shaped laboratory spaces like rectangular rooms. However, in consumer use-cases, the physical rooms often include static obstacles like furniture and may even contain other people/users. Consequently, practical VR play areas are non-convex and often dynamic. In such cases, content-unaware methods are highly likely to cause collisions, as seen by example in the supplementary video. In contrast, our dynamic technique and a real-time implementation can respond to physical-world changes, guiding users away from boundaries and obstacles.

To simulate multi-user scenarios, we use the recorded physical paths from IMAGE-SGD-I and II as moving obstacle positions, and then use our dynamic path planner to simulate new walking paths with their corresponding virtual space records as input. The dynamic planner reduces the error ϵ from obstacles by 94.2% on average ($SD = 3.9\%$). The overall average ξ is $2.82e{-}3$ ($SD = 1.82e{-}3$) for the simulation, which is lower than the original ($3.06e{-}3$). However, ANOVA did not show a significant main effect ($F_{1,17} = 1.055, p = 0.319$). This means that our method may introduce extra boundary errors by avoiding moving obstacles, but this is not statistically significant.

Figures 50b and 50c show additional simulated paths for redirection around static and dynamic obstacles. The supplemental video also contains a non-simulated example, where, since our current setup cannot track multiple users, the "moving obstacle" is just another person instructed to walk along a predetermined path.

Dynamic path planning versus static scene warping The static scene warping methods in [6, 40] depend on significant occlusions in the virtual scene to drive unnoticeable geometric warping. These methods can thus cause visible artifacts or scaling for open virtual spaces. Our dynamic path planning method can handle

step / platform	derivative	function	sampling
GPU	0.005	0.004	0.0045
CPU	0.025	0.016	0.74

Table 4: Performance comparison between our GPU-based path planner (Section 6.3.2) and a CPU implementation. *The GPU time consumption already includes memory transferring between GPU and CPU. The three parts are dynamic sampling (Figure 45), the computation of the cost function Equation* (71) *and its derivatives.*

both open and occluded virtual spaces, since it does not rely on any scene appearance. Figure 52 shows a comparison. Moreover, unlike Sun et al. [6] and Dong et al. [40], our planning approach runs in real-time, so it can also redirect the user to fit physical environmental changes.

6.5.5 Performance

Table 4 compares our GPU-based sampling and optimization with a corresponding CPU implementation. It shows that we are able to achieve a significant speedup compared to the CPU, enabling real-time dynamic path planning without latency. Combined with our amortization approach from Section 6.4.2, we are able to run our overall system including eye tracking, dynamic path planning, and rendering at 80-85 FPS depending on rendering complexity.

6.6 Applications

Beyond redirected walking with more perceptual comfort and visual quality, our system can benefit other applications:

Cinematic VR Although users can freely explore in virtual scenes, directors who produce immersive stories may intend to redirect the user to a certain part of the scene. Our path planning approach (Section 6.3.2) can adapt to story-based objectives to achieve this.

Home entertainment Our method lets multiple users explore the same or different virtual scenes while sharing one physical room. Home entertainment applications often contain multiple game players in the same room. It could encourage game industry development towards VR platform by avoiding unnatural locomotion controlling and enabling practical features such as inter-user collaboration and competition.

Education In architectural design education or virtual museum navigation scenarios, users should be guided to follow an ideal path to increase exposure or avoid getting lost. Our redirection approach can be adjusted to guide users towards predefined virtual paths.

6.7 Conclusion

In this work we showed that rotation-based redirection during saccades is effective in both room-scale and large-scale VR (Figure 53) and that our GPU implementation allows real-time path planning (Table 4). The real-time performance also allows the planning to avoid moving obstacles and changing geometry. However, recent researches on robotics and artificial intelligence fields may be adapted to the redirection planning approach for faster and more robust response to the dynamic environmental changes.

Limiting our redirection transformations to rotational gain simplified the planning optimization, enabling real-time performance. We plan to investigate whether translational gain can be incorporated into the optimization while maintaining real-time performance.

There are many opportunities for enhancing the redirection system, including saccade prediction [140,141] to compensate for tracking latency, learning [157] to enhance gaze guidance, redirection during blink suppression [131] to provide more opportunities for redirection, and additional forms of distractors [152, 153] to encourage more eye movement.

Compared to warping based methods [6, 40], our rigid-transformation based redirection allows exploring open virtual spaces without distracting visual distortions. However, Suma et al. [158] showed that warping provides chances to overlay virtual objects onto physical obstacles in applications like mixed reality. To further

enhance the visual and tactile consistency between the virtual and physical environments, we plan to investigate incorporating limited degrees of warping [159].

Our system works for room-scale physical environments. However, the space saving benefit from the saccadic gain increases greatly as the available physical area grows. Figure 53 shows the comparison and trend; it would be interesting to investigate whether further gains could come from tuning the system for larger areas.

7 Future Work

7.1 Eye-tracked Immersive Perception

Visual system movements happen much more frequently than other human actions such as head rotation and body movement. Thus, a critical factor to obtain perceptual information is by tracking, analyzing and leveraging the eye.

In addition to distorted visual stimuli [6], immersive locomotion in both VR and AR may also benefit from varied retinal nature. For example, the rod cells, who are sensitive to motions, have much denser distribution in the periphery, which is in contrast to the that of cone cells. Making the redirection and rendering adaptive to both content and the eye motion receptors may deliver better experience.

Compared to redirected walking, non-locomotive scenarios, such as large scale media focus+context visualization [160] or cinematography [161], require strong scene understanding perception. This includes color, depth or level-of-detail. Performing the rendering with respect to actual gaze position than display center (a common assumption, e.g., Hololens) provides attention-precise stimuli. In addition, there are other eye actions such as vergence, blink or saccade [141]. Considering these can all provide further information to make perception-guided experience.

7.2 Perception-Assisted Immersive Interface

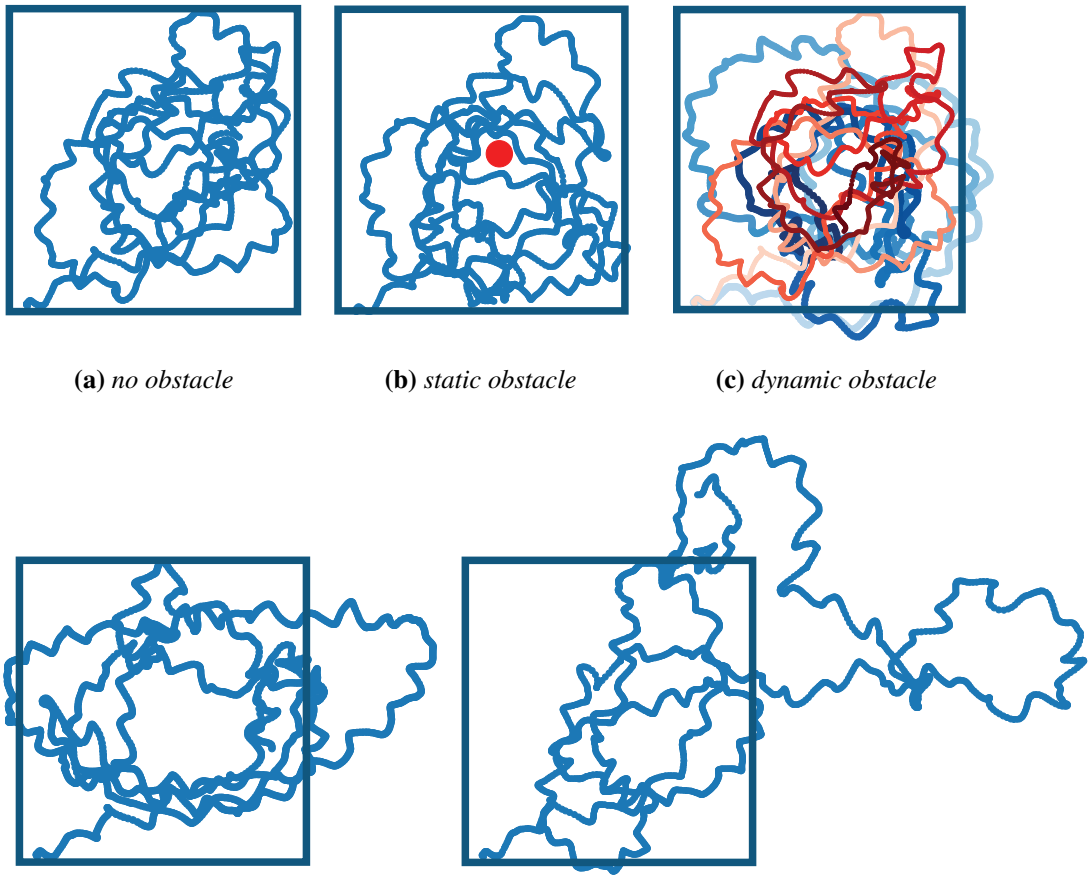
Interactive content creation/editing, such as painting, sculpting, modelling or animating are shown to benefit from VR/AR by a variety of applications. Interaction

requires accurate position understanding to both the user and system. However, current immersive applications lack enough position and depth cue support. As a result, users commonly experience inaccurate element positioning. For example, misplacement of brush curve or target destination. This causes unrealistic experience and thus limits broader productive mixed reality applications such as architectural/artistic design or CAD.

Hardware-wise, next generation display systems with real-time rendering [10], for example, light field or holographic displays, can be direct solutions to this problem. However, even with currently customer-level mono/stereo displays, a depth cue (from gaze and vergence) enhanced 2D rendering may also reduce the interaction misplacement. The future goal is to provide better immersive interaction experience guided by human perceptual factors, especially on better position/depth cues.

7.3 Longer Term

As a recent sci-fi film "Ready Player One" imagined, the future of VR/AR platform can potentially replace the current visual media carriers, such as projectors and monitors, for daily usage. However, as described in this thesis, those all require significant improvements in both software and hardware. Driven by the amount of user expectations and challenges, we predict that in the long run, the research community will keep exploring and see new discoveries in viewing comfort, room-scale locomotion and fast enough performance. The potential opportunities for computing and manipulating users' perception to the immersive media may not be limited to eye-tracking but even further steps, for example, brain signal tracking and multi-model feedbacks including haptics and sounds.



(a) *no obstacle* **(b)** *static obstacle* **(c)** *dynamic obstacle*

(d) *no obstacle, head-only* **(e)** *no obstacle, S2C (steer to center)*

Figure 50: Path planning comparisons under different types of obstacles and redirection methods. *All experiments share the same physical room (visualized as the dark blue rectangles). Using a randomly sampled user's historical virtual head and gaze data from Figure 43c, we simulate our dynamic path planning with no obstacle in (a), with a static obstacle (the red circle) in (b), with a dynamic obstacle (the red curve) in (c), using head-rotation without saccade in (d), and using traditional S2C redirection in (e). Both (d) and (e) have no obstacle and can be compared to (a). All but (a) are simulated paths. The saturation gradients in (c) stand for time propagation.*

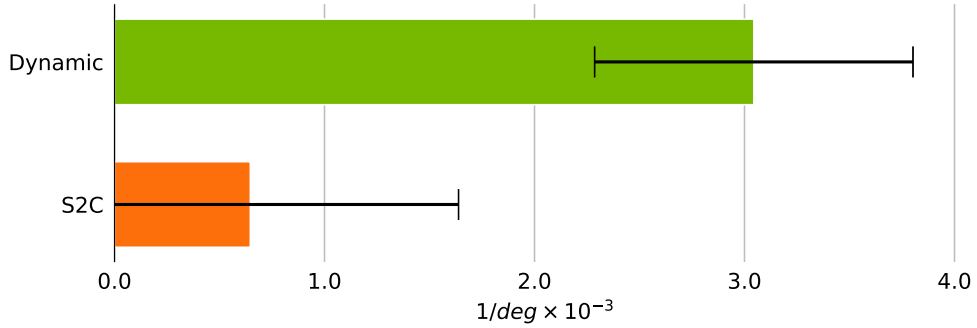
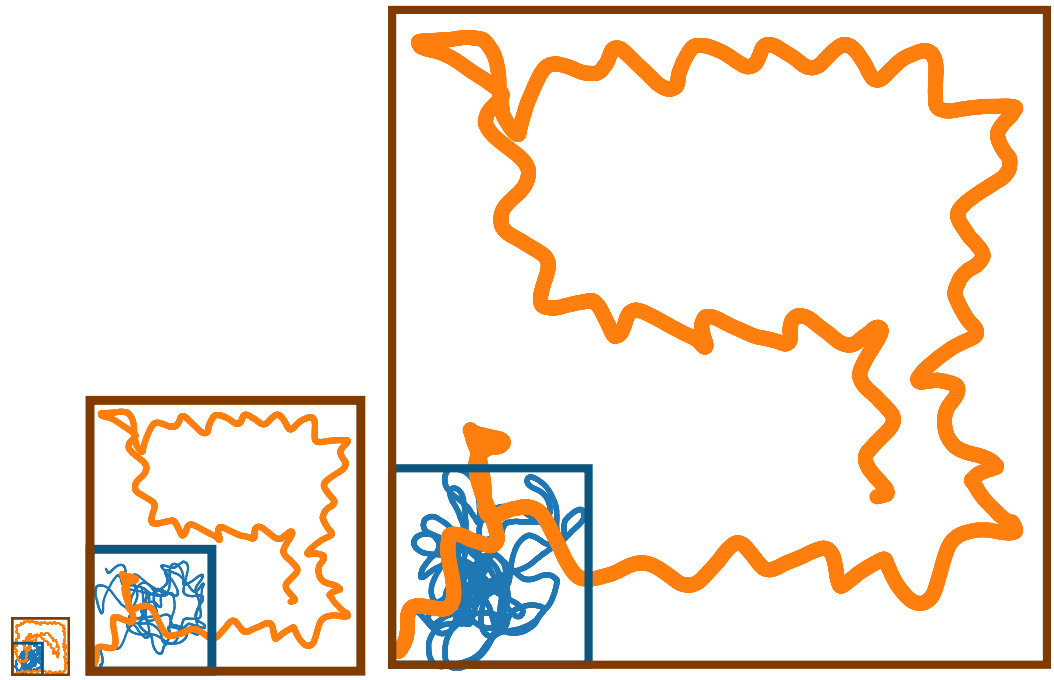


Figure 51: The effectiveness of dynamic path planning. *Here we show the average saving ratios ξ and 95% confidence intervals over all trials from IMAGE-SGD-I and IMAGE-SGD-II in Section 6.5.2, and corresponding simulated S2C redirection with identical input virtual path. It can be seen that the redirection with our dynamic planing approach shows stronger error saving than S2C redirection (both with saccadic redirection and SGD).*

(a) *static warping* **(b)** *dynamic path planning*

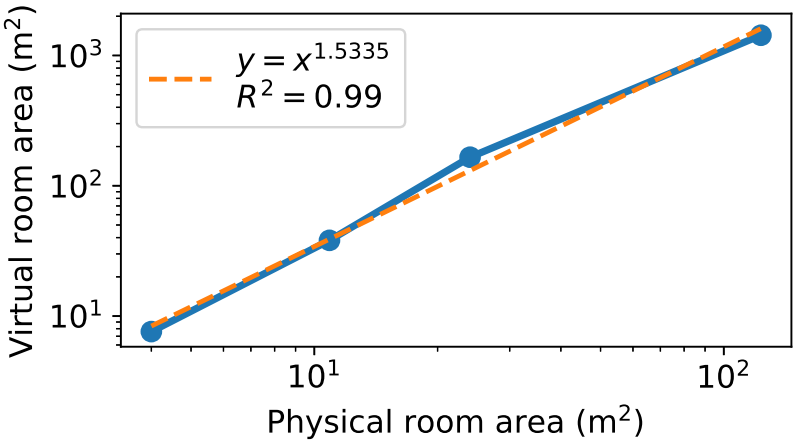
Figure 52: Static scene warping versus dynamic path planning. *Prior static warping methods such as [6, 40] rely on sufficient occlusions and may cause unnatural distortions or translation/rotation gains for sufficiently open spaces as shown in (a). Our method, in contrast, does not cause scene distortions or noticeable translation/rotation gains (b). Scene courtesy of Tirgames.*



(a) *tiny*

(b) *medium*

(c) *large*



(d) *trend*

Figure 53: Simulation results of different physical spaces. *(a) shows the result in Figure 43c with* $12.25\,\mathrm{m}^2$ *physical and* $40.96\,\mathrm{m}^2$ *virtual spaces. (b) shows a simulation result for a physical room of* $47.61\,\mathrm{m}^2$*. The simulated user can walk through a virtual space of 4.9× larger area. (c) shows a simulation result for a physical room of* $134.56\,\mathrm{m}^2$*. The virtual space can be 11.1× larger in area. As a comparison, in the user experiments Section 6.5.2, the physical room size was* $3.61\,\mathrm{m}^2$*, the virtual space can be 2.0× in area. The trend is plotted in Figure 53d.*

References

[1] Koosha Mirhosseini, Qi Sun, Krishna C Gurijala, Bireswar Laha, and Arie E Kaufman. Benefits of 3d immersion for virtual colonoscopy. In *3DVis (3DVis), 2014 IEEE VIS International Workshop on*, pages 75–79. IEEE, 2014.

[2] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *CVPR '15*, pages 5556–5565, 2015.

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

[4] Stéphane Bouchard, Geneviève Robillard, and Patrice Renaud. Revising the factor structure of the simulator sickness questionnaire. *Annual Review of CyberTherapy and Telemedicine*, 5:128–137, 2007.

[5] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks, Jr. Walking $>$ walking-in-place $>$ flying, in virtual environments. In *SIGGRAPH '99*, pages 359–364, 1999.

[6] Qi Sun, Li-Yi Wei, and Arie Kaufman. Mapping virtual and physical reality. *ACM Trans. Graph.*, 35(4), 2018.

[7] Joohwan Kim, Qi Sun, Fu-Chung Huang, Li-Yi Wei, David Luebke, and Arie Kaufman. Perceptual studies for foveated light field displays. *CoRR*, abs/1708.06034, 2017.

[8] Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein. The light field stereoscope: Immersive computer graphics via factored near-eye light field displays with focus cues. *ACM Trans. Graph.*, 34(4):60:1–60:12, July 2015.

[9] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph.*, 35(6):179:1–179:12, November 2016.

[10] Qi Sun, Fu-Chung Huang, Joohwan Kim, Li-Yi Wei, David Luebke, and Arie Kaufman. Perceptually-guided foveation for light field displays. *ACM Trans. Graph.*, 36(6), November 2017.

[11] Qi Sun, Seyedkoosha Mirhosseini, Ievgeniia Gutenko, Ji Hwan Park, Charilaos Papadopoulos, Bireswar Laha, and Arie Kaufman. Buyers satisfaction in a virtual fitting room scenario based on realism of avatar. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 183–184. IEEE, 2015.

[12] Lichan Hong, Shigeru Muraki, Arie Kaufman, Dirk Bartz, and Taosong He. Virtual voyage: Interactive navigation in the human colon. In *SIGGRAPH '97*, pages 27–34, 1997.

[13] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper. Virtual Environ.*, 7(3):225–240, June 1998.

[14] Sharif Razzaque, David Swapp, Mel Slater, Mary C. Whitton, and Anthony Steed. Redirected walking in place. In *EGVE '02*, pages 123–130, 2002.

[15] Eric Hodgson, Eric Bachmann, and David Waller. Redirected walking to explore virtual environments: Assessing the potential for spatial interference. *ACM Trans. Appl. Percept.*, 8(4):22:1–22:22, December 2008.

[16] E.A. Suma, Z. Lipps, S. Finkelstein, D.M. Krum, and M. Bolas. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):555–564, April 2012.

[17] K. Vasylevska, H. Kaufmann, M. Bolas, and E.A. Suma. Flexible spaces: Dynamic layout generation for infinite walking in virtual environments. In *3DUI '13*, pages 39–42, March 2013.

[18] Lung-Pan Cheng, Thijs Roumen, Hannes Rantzsch, Sven Köhler, Patrick Schmidt, Robert Kovacs, Johannes Jasper, Jonas Kemper, and Patrick Baudisch. Turkdeck: Physical virtual reality based on people. In *UIST '15*, pages 417–426, 2015.

[19] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. Analyses of human sensitivity to redirected walking. In *VRST '08*, pages 149–156, 2008.

[20] N.C. Nilsson, S. Serafin, and R. Nordahl. Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):569–578, April 2014.

[21] G. Bruder, P. Lubas, and F. Steinicke. Cognitive resource demands of redirected walking. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):539–544, April 2015.

[22] Roi Poranne and Yaron Lipman. Provably good planar mappings. *ACM Trans. Graph.*, 33(4):76:1–76:11, 2014.

[23] Xiao-Ming Fu, Yang Liu, and Baining Guo. Computing locally injective mappings by advanced mips. *ACM Trans. Graph.*, 34(4):71:1–71:12, July 2015.

[24] Renjie Chen and Ofir Weber. Bounded distortion harmonic mappings in the plane. *ACM Trans. Graph.*, 34(4):73:1–73:12, July 2015.

[25] Leonard McMillan, Jr. *An Image-based Approach to Three-dimensional Computer Graphics*. PhD thesis, 1997. UMI Order No. GAX97-30561.

[26] Lei Yang, Yu-Chiu Tse, Pedro V. Sander, Jason Lawrence, Diego Nehab, Hugues Hoppe, and Clara L. Wilkins. Image-based bidirectional scene reprojection. *ACM Trans. Graph.*, 30(6):150:1–150:10, December 2011.

[27] Voicu Popescu, Paul Rosen, and Nicoletta Adamo-Villani. The graph camera. *ACM Trans. Graph.*, 28(5):158:1–158:8, December 2009.

[28] K. Carnegie and T. Rhee. Reducing visual discomfort with hmds using dynamic depth of field. *IEEE Computer Graphics and Applications*, 35(5):34–41, Sept 2015.

[29] Andrew Maimone, Gordon Wetzstein, Matthew Hirsch, Douglas Lanman, Ramesh Raskar, and Henry Fuchs. Focus 3d: Compressive accommodation display. *ACM Trans. Graph.*, 32(5):153:1–153:13, October 2013.

[30] Andrew Maimone and Henry Fuchs. Computational augmented reality eyeglasses. In *ISMAR '13*, pages 29–38, 2013.

[31] Douglas Lanman and David Luebke. Near-eye light field displays. *ACM Trans. Graph.*, 32(6):220:1–220:10, November 2013.

[32] Rahul Narain, Rachel A. Albert, Abdullah Bulbul, Gregory J. Ward, Martin S. Banks, and James F. O'Brien. Optimal presentation of imagery with focus cues on multi-plane displays. *ACM Trans. Graph.*, 34(4):59:1–59:12, July 2015.

[33] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3d graphics. *ACM Trans. Graph.*, 31(6):164:1–164:10, November 2012.

[34] Y. Takaki. High-density directional display for generating natural three-dimensional images. *Proceedings of the IEEE*, 94(3):654–663, March 2006.

[35] Yasuhiro Takaki, Kosuke Tanaka, and Junya Nakamura. Super multi-view display with a lower resolution flat-panel display. *Opt. Express*, 19(5):4129–4139, Feb 2011.

[36] Vitor F. Pamplona, Manuel M. Oliveira, Daniel G. Aliaga, and Ramesh Raskar. Tailored displays to compensate for visual aberrations. *ACM Trans. Graph.*, 31(4):81:1–81:12, July 2012.

[37] Fu-Chung Huang, Gordon Wetzstein, Brian A. Barsky, and Ramesh Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Trans. Graph.*, 33(4):59:1–59:12, July 2014.

[38] Sharif Razzaque, Zachariah Kohn, and Mary C. Whitton. Redirected Walking. In *Eurographics 2001 - Short Presentations*. Eurographics Association, 2001.

[39] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27, Jan 2010.

[40] Zhi-Chao Dong, Xiao-Ming Fu, Chi Zhang, Kang Wu, and Ligang Liu. Smooth assembled mappings for large-scale real walking. *ACM Trans. Graph.*, 36(6):211:1–211:13, November 2017.

[41] Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance. In *Eurographics Symposium on Virtual Environments (2015)*, pages 93–100, Kyoto, Japan, October 2015. The Eurographics Association.

[42] J.Johanna Hopp and Albert F Fuchs. The characteristics and neuronal substrate of saccadic eye movement plasticity. *Progress in Neurobiology*, 72(1):27 – 53, 2004.

[43] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. Facial performance sensing head-mounted display. *ACM Trans. Graph.*, 34(4):47:1–47:9, July 2015.

[44] Youngkyoon Jang, Seung-Tak Noh, Hyung Jin Chang, Tae-Kyun Kim, and Woontack Woo. 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):501–510, April 2015.

[45] J. L. Souman, P. Robuffo Giordano, M. Schwaiger, I. Frissen, T. Thümmel, H. Ulbrich, A. De Luca, H. H. Bülthoff, and M. O. Ernst. Cyberwalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Trans. Appl. Percept.*, 8(4):25:1–25:22, December 2008.

[46] Hiroo Iwata, Hiroaki Yano, and Hiroshi Tomioka. Powered shoes. In *SIGGRAPH '06 Emerging Technologies*, 2006.

[47] Martin Schwaiger, Thomas Thümmel, and Heinz Ulbrich. Cyberwalk: Implementation of a ball bearing platform for humans. In *Human-Computer Interaction. Interaction Platforms and Techniques*, pages 926–935. Springer, 2007.

[48] Raymond Wong. Vr startups: Stop trying to make virtual reality treadmills a thing, 2015. http://mashable.com/2015/06/20/virtual-reality-treadmills/.

[49] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. Substitutional reality: Using the physical environment to design virtual reality experiences. In *CHI '15*, pages 3307–3316, 2015.

[50] Steven Maesen, Patrik Goorts, and Philippe Bekaert. Scalable optical tracking for navigating large virtual environments using spatially encoded markers. In *VRST '13*, pages 101–110, 2013.

[51] M.A. Zmuda, J.L. Wonser, E.R. Bachmann, and E. Hodgson. Optimizing constrained-environment redirected walking instructions using search techniques. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1872–1884, Nov 2013.

[52] T. Nescher, Ying-Yin Huang, and A. Kunz. Planning redirection techniques for optimal free walking experience using model predictive control. In *3DUI '14*, pages 111–118, March 2014.

[53] Ruimin Zhang and Scott A. Kuhl. Human sensitivity to dynamic rotation gains in head-mounted displays. In *SAP '13*, pages 71–74, 2013.

[54] Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. Locally injective mappings. In *SGP '13*, pages 125–135, 2013.

[55] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, July 2002.

[56] Alfred Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, Inc., 1st edition, 1996.

[57] Ran Gal, Olga Sorkine, and Daniel Cohen-Or. Feature-aware texturing. In *EGSR '06*, pages 297–303, 2006.

[58] J. Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects (Universitext)*. Springer-Verlag, 2006.

[59] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[60] Cyril Crassin, Morgan McGuire, Kayvon Fatahalian, and Aaron Lefohn. Aggregate G-buffer anti-aliasing. In *I3D '15*, pages 109–119, 2015.

[61] Greg Turk and James F. O'Brien. Shape transformation using variational implicit functions. In *SIGGRAPH '05 Courses*, 2005.

[62] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH '96*, pages 11–20, 1996.

[63] Doug A. Bowman, Joseph L. Gabbard, and Deborah Hix. A survey of usability evaluation in virtual environments: Classification and comparison of methods. *Presence: Teleoper. Virtual Environ.*, 11(4):404–424, August 2002.

[64] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[65] Jonas Schild, Joseph LaViola, and Maic Masuch. Understanding user experience in stereoscopic 3d games. In *CHI '12*, pages 89–98, 2012.

[66] Jian Cui, Paul Rosen, Voicu Popescu, and Christoph Hoffmann. A curved ray camera for handling occlusions through continuous multiperspective visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1235–1242, November 2010.

[67] Bin Wang, Kenneth J. Ciuffreda, and Trevor Irish. Equiblur zones at the fovea and near retinal periphery. *Vision Research*, 46(21):3690 – 3698, 2006.

[68] Hugh R Taylor. Applying new design principles to the construction of an illiterate e chart. *Optometry & Vision Science*, 55(5):348–351, 1978.

[69] HCCH Levitt. Transformed up-down methods in psychoacoustics. *The Journal of the Acoustical society of America*, 49(2B):467–477, 1971.

[70] Heiko H. Schütt, Stefan Harmeling, Jakob H. Macke, and Felix A. Wichmann. Painfree and accurate bayesian estimation of psychometric functions for (potentially) overdispersed data. *Vision Research*, 122:105–123, 2016.

[71] Anne Seidemann, Frank Schaeffel, Antonio Guirao, Noberto Lopez-Gil, and Pablo Artal. Peripheral refractive errors in myopic, emmetropic, and hyperopic young subjects. *JOSA A*, 19(12):2363–2373, 2002.

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

[73] Anjul Patney, Marina Zannoli, George-Alex Koulieris, Joohwan Kim, Gordon Wetzstein, and Frank Steinicke. Applications of visual perception to virtual reality rendering. In *SIGGRAPH '17 Courses*, pages 1:1–1:38, 2017.

[74] Kurt Akeley, Simon J. Watt, Ahna Reza Girshick, and Martin S. Banks. A stereo display prototype with multiple focal distances. *ACM Trans. Graph.*, 23(3):804–813, August 2004.

[75] Gordon Wetzstein, Douglas Lanman, Wolfgang Heidrich, and Ramesh Raskar. Layered 3d: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. *ACM Trans. Graph.*, 30(4):95:1–95:12, July 2011.

[76] Gordon Wetzstein, Douglas Lanman, Matthew Hirsch, and Ramesh Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph.*, 31(4):80:1–80:11, July 2012.

[77] Matias Koskela, Timo Viitanen, Pekka Jääskeläinen, and Jarmo Takala. Foveated path tracing. In *ISVC '16*, pages 723–732, 2016.

[78] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *SIGGRAPH '00*, pages 307–318, 2000.

[79] Ren Ng. Fourier slice photography. *ACM Trans. Graph.*, 24(3):735–744, July 2005.

[80] Anat Levin, Samuel W. Hasinoff, Paul Green, Frédo Durand, and William T. Freeman. 4d frequency analysis of computational cameras for depth of field extension. *ACM Trans. Graph.*, 28(3):97:1–97:14, July 2009.

[81] V. Ramachandra, K. Hirakawa, M. Zwicker, and T. Nguyen. Spatioangular prefiltering for multiview 3d displays. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):642–654, May 2011.

[82] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96*, pages 31–42, 1996.

[83] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96*, pages 43–54, 1996.

[84] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.*, 27(3):33:1–33:10, August 2008.

[85] Kevin Egan, Yu-Ting Tseng, Nicolas Holzschuch, Frédo Durand, and Ravi Ramamoorthi. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.*, 28(3):93:1–93:13, July 2009.

[86] Kevin Egan, Florian Hecht, Frédo Durand, and Ravi Ramamoorthi. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph.*, 30(2):9:1–9:13, April 2011.

[87] Kevin Egan, Frédo Durand, and Ravi Ramamoorthi. Practical filtering for efficient ray-traced directional occlusion. *ACM Trans. Graph.*, 30(6):180:1–180:10, December 2011.

[88] Jaakko Lehtinen, Timo Aila, Jiawen Chen, Samuli Laine, and Frédo Durand. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.*, 30(4):55:1–55:12, July 2011.

[89] Ling-Qi Yan, Soham Uday Mehta, Ravi Ramamoorthi, and Fredo Durand. Fast 4d sheared filtering for interactive rendering of distribution effects. *ACM Trans. Graph.*, 35(1):7:1–7:13, December 2015.

[90] Alexander Wender, Julian Iseringhausen, Bastian Goldluecke, Martin Fuchs, and Matthias B. Hullin. Light field imaging through household optics. In *Vision, Modeling & Visualization*, 2015.

[91] Li-Yi Wei, Chia-Kai Liang, Graham Myhre, Colvin Pitts, and Kurt Akeley. Improving light field camera sample design with irregularity and aberration. *ACM Trans. Graph.*, 34(4):152:1–152:11, July 2015.

[92] Julian Iseringhausen, Bastian Goldlücke, Nina Pesheva, Stanimir Iliev, Alexander Wender, Martin Fuchs, and Matthias B. Hullin. 4d imaging through spray-on optics. *ACM Trans. Graph.*, 36(4):35:1–35:11, July 2017.

[93] Donald G Dansereau, Glenn Schuster, Joseph Ford, and Gordon Wetzstein. A wide-field-of-view monocentric light field camera. In *CVPR '17*, 2017.

[94] Jaakko Lehtinen, Timo Aila, Samuli Laine, and Frédo Durand. Reconstructing the indirect light field for global illumination. *ACM Trans. Graph.*, 31(4):51:1–51:10, July 2012.

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

[96] Neville Drasdo, C. Leigh Millican, Charles R. Katholi, and Christine A. Curcio. The length of henle fibers in the human retina and a model of ganglion receptive field density in the visual field. *Vision Research*, 47(22):2901 – 2911, 2007.

[97] LN Thibos, FE Cheney, and DJ Walsh. Retinal limits to the detection and resolution of gratings. *JOSA A*, 4(8):1524–1529, 1987.

[98] Matthias Zwicker, Wojciech Matusik, Frédo Durand, Hanspeter Pfister, and Clifton Forlines. Antialiasing for automultiscopic 3d displays. In *SIGGRAPH '06 Sketches*, 2006.

[99] WN Charman and J Tucker. Accommodation as a function of object form. *Optometry & Vision Science*, 55(2):84–92, 1978.

[100] Marina Zannoli, Gordon D. Love, Rahul Narain, and Martin S. Banks. Blur and the perception of depth at occlusions. *Journal of Vision*, 16(6):17, 2016.

[101] Jonathan Ragan-Kelley, Jaakko Lehtinen, Jiawen Chen, Michael Doggett, and Frédo Durand. Decoupled sampling for graphics pipelines. *ACM Trans. Graph.*, 30(3):17:1–17:17, May 2011.

[102] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *UbiComp '14 Adjunct*, pages 1151–1160, 2014.

[103] Thomas S. A. Wallis, Matthias Bethge, and Felix A. Wichmann. Testing models of peripheral encoding using metamerism in an oddity paradigm. *Journal of Vision*, 16(2):4, 2016.

[104] John Siderov and Ronald S Harwerth. Stereopsis, spatial frequency and retinal eccentricity. *Vision research*, 35(16):2329–2337, 1995.

[105] Lester C. Loschky and Gary S. Wolverton. How late can you update gaze-contingent multiresolutional displays without detection? *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(4):7:1–7:10, December 2007.

[106] K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, T. Janczak, and A. Lefohn. Coarse pixel shading. In *HPG '14*, pages 9–18, 2014.

[107] Felix Heide, Gordon Wetzstein, Ramesh Raskar, and Wolfgang Heidrich. Adaptive image synthesis for compressive displays. *ACM Trans. Graph.*, 32(4):132:1–132:12, July 2013.

[108] Soham Mehta, Brandon Wang, and Ravi Ramamoorthi. Axis-aligned filtering for interactive sampled soft shadows. *ACM Trans. Graph.*, 31(6):163:1–163:10, Nov 2012.

[109] Christopher W Tyler. Analysis of visual modulation sensitivity. iii. meridional variations in peripheral flicker sensitivity. *JOSA A*, 4(8):1612–1619, 1987.

[110] Greg Egan. *Permutation City*. Millennium Orion Publishing Group, 1994.

[111] Yuyang Qiu and Ling Zhu. The best approximation of the sinc function by a polynomial of degree with the square norm. *Journal of Inequalities and Applications*, 2010(1):1–12, 2010.

[112] Sharif Razzaque. *Redirected Walking*. PhD thesis, Chapel Hill, NC, USA, 2005. AAI3190299.

[113] Eric Hodgson and Eric Bachmann. Comparing four approaches to generalized redirected walking: Simulation and live user data. *TVCG*, 19(4):634–643, April 2013.

[114] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *CHI '16*, 2016.

[115] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D. Wilson. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3718–3728, New York, NY, USA, 2017. ACM.

[116] Mahdi Azmandian, Timofey Grechkin, and Evan Suma Rosenberg. An evaluation of strategies for two-user redirected walking in shared physical spaces. In *2017 IEEE Virtual Reality (VR)*, pages 91–98, March 2017.

[117] Stephen M. Reder. On-line monitoring of eye-position signals in contingent and noncontingent paradigms. *Behavior Research Methods*, 5(2):218–228, 1973.

[118] Andrew T. Duchowski, Nathan Cournia, and Hunter Murphy. Gaze-contingent displays: A review. *CyberPsychology & Behavior*, 7(6):621–634, 2004.

[119] Peter Vincent and Ritchie Brannan. S7797 Tobii Eye Tracked Foveated Rendering for VR and Desktop, May 2017.

[120] Andrew T. Duchowski and Arzu Çöltekin. Foveated gaze-contingent displays for peripheral LOD management, 3d visualization, and stereo imaging. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(4):6, 2007.

[121] Marc Levoy and Ross Whitaker. Gaze-directed volume rendering. *ACM SIGGRAPH Computer Graphics*, 24(2):217–223, 1990.

[122] David Luebke and Benjamin Hallen. Perceptually driven simplification for interactive rendering. In *Rendering Techniques 2001*, pages 223–234. Springer, 2001.

[123] Anjul Patney. Perceptual insights into foveated virtual reality. In *NVIDIA GPU Technology Conference 2017 Talks*, San Jose, California, May 2017.

[124] Rachel Albert, Anjul Patney, David Luebke, and Joohwan Kim. Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.*, 14(4):25:1–25:13, September 2017.

[125] David E. Jacobs, Orazio Gallo, Emily A. Cooper, Kari Pulli, and Marc Levoy. Simulating the visual experience of very bright and very dark scenes. *ACM Trans. Graph.*, 34(3):25:1–25:15, May 2015.

[126] Andrew T. Duchowski, Donald H. House, Jordan Gestring, Rui I. Wang, Krzysztof Krejtz, Izabela Krejtz, Radosław Mantiuk, and Bartosz Bazyluk. Reducing visual discomfort of 3d stereoscopic displays with gaze-contingent depth-of-field. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '14, pages 39–46, New York, NY, USA, 2014. ACM.

[127] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. Gaze + pinch interaction in virtual reality. In *Proc. of Spatial User Interaction*, SUI '17, pages 99–108, 2017.

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

[129] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, 49(2):616–637, Apr 2017.

[130] Mounia Ziat, Vincent Hayward, C.Ẽlaine Chapman, Marc O. Ernst, and Charles Lenay. Tactile suppression of displacement. *Experimental Brain Research*, 206(3):299–310, Oct 2010.

[131] William H. Ridder III and Alan Tomlinson. A comparison of saccadic and blink suppression in normal observers. *Vision Research*, 37(22):3171–3179, November 1997.

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

[133] Mark R. Diamond, John Ross, and Maria C. Morrone. Extraretinal control of saccadic suppression. *The Journal of Neuroscience*, 20(9):3449–3455, 2000.

[134] Michael R. Ibbotson and Shaun L. Cloherty. Visual Perception: Saccadic Omission - Suppression or Temporal Masking? *Current Biology*, 19(12):R493–R496, June 2009.

[135] Ethel Matin. Saccadic suppression: a review and an analysis. *Psychological bulletin*, 81(12):899, 1974.

[136] George W. McConkie and Lester C. Loschky. Perception onset time during fixations in free viewing. *Behavior Research Methods, Instruments, & Computers*, 34(4):481–490, 2002.

[137] John Ross, M. Concetta Morrone, Michael E. Goldberg, and David C. Burr. Changes in visual perception at the time of saccades. *Trends in Neurosciences*, 24(2):113–121, February 2001.

[138] Bruce Bridgeman, Derek Hendry, and Lawrence Stark. Failure to detect displacement of the visual world during saccadic eye movements. *Vision Research*, 15(6):719 – 722, 1975.

[139] Wenxun Li and Leonard Matin. The influence of saccade length on the saccadic suppression of displacement detection. *Perception & Psychophysics*, 48(5):453–458, Sep 1990.

[140] Peng Han, Daniel R Saunders, Russell L Woods, and Gang Luo. Trajectory prediction of saccadic eye movements using a compressed exponential model. *Journal of vision*, 13(8):27–27, 2013.

[141] Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4), 2017.

[142] Benjamin Bolte and Markus Lappe. Subliminal reorientation and repositioning in immersive virtual environments using saccadic suppression. *IEEE TVCG*, 21(4):545–552, April 2015.

[143] Eike Langbehn, Gerd Bruder, and Frank Steinicke. Subliminal reorientation and repositioning in virtual reality during eye blinks. In *Proc. of Spatial User Interaction*, pages 213–213, 2016.

[144] Reynold Bailey, Ann McNamara, Nisha Sudarsanam, and Cindy Grimm. Subtle gaze direction. *ACM Trans. Graph.*, 28(4):100:1–100:14, September 2009.

[145] Ann McNamara, Reynold Bailey, and Cindy Grimm. Improving search task performance using subtle gaze direction. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, APGV '08, pages 51–56, New York, NY, USA, 2008. ACM.

[146] Ann McNamara, Reynold Bailey, and Cindy Grimm. Search task performance using subtle gaze direction with the presence of distractions. *ACM Trans. Appl. Percept.*, 6(3):17:1–17:19, September 2009.

[147] Ann McNamara, Thomas Booth, Srinivas Sridharan, Stephen Caffey, Cindy Grimm, and Reynold Bailey. Directing gaze in narrative art. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '12, pages 63–70, New York, NY, USA, 2012. ACM.

[148] Steve Grogorick, Michael Stengel, Elmar Eisemann, and Marcus Magnor. Subtle gaze guidance for immersive environments. In *SAP '17*, pages 4:1–4:7, 2017.

[149] Srinivas Sridharan, James Pieszala, and Reynold Bailey. Depth-based subtle gaze guidance in virtual reality environments. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, SAP '15, pages 132–132, New York, NY, USA, 2015. ACM.

[150] Timofey Grechkin, Jerald Thomas, Mahdi Azmandian, Mark Bolas, and Evan Suma. Revisiting detection thresholds for redirected walking: Com-

bining translation and curvature gains. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, pages 113–120, New York, NY, USA, 2016. ACM.

[151] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[152] Haiwei Chen and Henry Fuchs. Supporting free walking in a large virtual environment: Imperceptible redirected walking with an immersive distractor. In *Proceedings of the Computer Graphics International Conference*, CGI '17, pages 22:1–22:6, New York, NY, USA, 2017. ACM.

[153] Tabitha C Peck, Henry Fuchs, and Mary C Whitton. Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *Virtual Reality Conference*, pages 35–38. IEEE, 2010.

[154] Srinivas Sridharan and Reynold Bailey. Automatic target prediction and subtle gaze guidance for improved spatial information recall. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, SAP '15, pages 99–106, 2015.

[155] Thomas Booth, Srinivas Sridharan, Ann McNamara, Cindy Grimm, and Reynold Bailey. Guiding attention in controlled real-world environments. In *SAP '13*, pages 75–82, 2013.

[156] Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pages 9–14, March 2016.

[157] Leon A. Gatys, Matthias Kümmerer, Thomas Wallis, and Matthias Bethge. Guiding human gaze with convolutional neural networks. *ArXiv e-prints*, December 2017.

[158] Evan A. Suma, D. Krum, and Mark Bolas. Redirected walking in mixed reality training applications. In *Human Walking in Virtual Environments: Perception, Technology, and Applications*, pages 319–331. Springer, 2013.

[159] Mahdi Azmandian, Timofey Grechkin, Mark Bolas, and Evan Suma. Automated path prediction for redirected walking using navigation meshes. In *IEEE Symposium on 3D User Interfaces*, pages 63–66, 2016.

[160] Huiwen Chang and Michael Cohen. Panning and zooming high-resolution panoramas in virtual reality devices. *UIST*, October 2017.

[161] Petr Kellnhofer, Piotr Didyk, Szu-Po Wang, Pitchaya Sitthi-Amorn, William Freeman, Fredo Durand, and Wojciech Matusik. 3DTV at home: Eulerian-lagrangian stereo-to-multiview conversion. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4), 2017.