Reducing Simulator Sickness with Perceptual Camera Control

PING HU, Stony Brook University
QI SUN, Adobe Research
PIOTR DIDYK, Universitá della Svizzera italiana
LI-YI WEI, Adobe Research
ARIE E. KAUFMAN, Stony Brook University

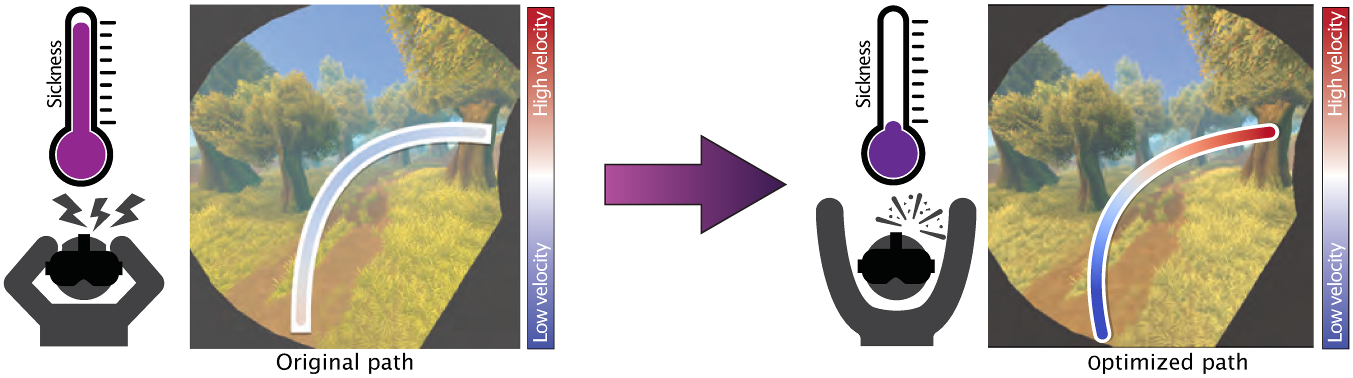


Fig. 1. *Our method predicts visual discomfort in virtual navigation and optimizes a predefined view path to reduce simulator sickness.* In this example, the original path has a uniform animation speed. Our method automatically slows down around turns and speeds up on straight segments, leading to reduced motion sickness while maintaining the original design intention.

Virtual-reality provides an immersive environment but can induce cybersickness due to the discrepancy between visual and vestibular cues. To avoid this problem, the movement of the virtual camera needs to match the motion of the user in the real world. Unfortunately, this is usually difficult due to the mismatch between the size of the virtual environments and the space available to the users in the physical domain. The resulting constraints on the camera movement significantly hamper the adoption of virtual-reality headsets in many scenarios and make the design of the virtual environments very challenging. In this work, we study how the characteristics of the virtual camera movement (e.g., translational acceleration and rotational velocity) and the composition of the virtual environment (e.g., scene depth) contribute to perceived discomfort. Based on the results from our user experiments, we devise a computational model for predicting the magnitude of the discomfort for a given scene and camera trajectory. We further apply our model to a new path planning method which optimizes the input motion trajectory to reduce perceptual sickness. We evaluate the effectiveness of our method in improving perceptual comfort in a series of user studies targeting different applications. The results indicate that our method can reduce the perceived discomfort while maintaining the fidelity of the original navigation, and perform better than simpler alternatives.

CCS Concepts: • **Human-centered computing → Virtual reality**; • **Computing methodologies → Perception**.

Authors' addresses: Ping Hu, Stony Brook University; Qi Sun, Adobe Research; Piotr Didyk, Universitá della Svizzera italiana; Li-Yi Wei, Adobe Research; Arie E. Kaufman, Stony Brook University.

Additional Key Words and Phrases: virtual reality, head-mounted display, human perception

ACM Reference Format:
Ping Hu, Qi Sun, Piotr Didyk, Li-Yi Wei, and Arie E. Kaufman. 2019. Reducing Simulator Sickness with Perceptual Camera Control. *ACM Trans. Graph.* 38, 6, Article 210 (November 2019), 15 pages. https://doi.org/10.1145/3355089.3356490

1 INTRODUCTION

Virtual reality (VR) applications facilitate immersion, interaction, and sense of presence. While the freedom in stimulating the human sensory system using VR head-mounted displays opens great opportunities for exploring and interacting with the digital world, it may induce discomfort due to cybersickness [McCauley and Sharkey 1992]. When immersed in a virtual environment, the human visual system receives visual cues solely from the virtual world. At the same time, the vestibular system senses the physical movement of the observer in the real world. This decoupling often leads to inconsistencies between the visual and vestibular cues. A typical example is a situation when a real-world observer is seated and explores the virtual environment by moving along a complex trajectory. In such a case, the inconsistencies between the cues may introduce discomfort [Reason and Brand 1975]. The visual-vestibular conflict (VVC) is common in VR due to the widespread presence of extreme motions, including self-acceleration and self-rotation, in applications such as action games or flight simulators. Thus, reducing discomfort is critical for the usability of VR technology.

Despite many studies conducted to investigate the VVC and other sources of cybersickness as well as the growing concerns about the usage of head-mounted displays [Rebenitsch and Owen 2016], prevention or reduction of the effect still poses many challenges. One solution to reduce the VVC is to limit the virtual camera motion to movements that are performed in the real world. This, however, quickly becomes impractical due to space requirements. Redirected walking [Dong et al. 2017; Langbehn et al. 2018; Razzaque et al. 2002; Sun et al. 2018, 2016] addresses this problem, but is constrained by the currently available hardware, such as tethered headsets and stationary tracking cameras. Under stationary physical conditions, simulator sickness can be reduced on a per-frame basis, for example, by introducing independent visual background [Prothero et al. 1999], manipulating peripheral vision [Yao et al. 2014], or changing the field of view [Fernandes and Feiner 2016]. Unfortunately, these methods can also reduce the sense of presence.

In this work, we take a different approach and address the problem for the common stationary VR usage. Our goal is to alter the camera trajectory such that it provides a more comfortable viewing experience. The problem is related to camera control which has been extensively studied for graphics applications [Bell et al. 2001; Christie et al. 2008; He et al. 1996]; however, maintaining perceptual comfort remains an unexplored challenge. Motivated by physiology research, vision science, and many techniques that exploit perception to improve display quality [Masia et al. 2013; Weier et al. 2017], we design a series of experiments to validate the visual factors inducing visually induced motion sickness (VIMS), and construct a quantitative perceptual model to estimate VIMS. Based on this model, we further propose a path optimization method for automatic virtual camera control to reduce simulator sickness while maintaining the original design intentions. Our perceptual model and camera control can respond to dynamic scenes and users' head movements in each frame. We evaluate our perceptual model and camera control with a variety of applications (e.g., space flight, car racing, fighter simulator, colonoscopy) and usage scenarios (e.g., passive animation and active navigation), and demonstrate better perceptual comfort and intention matching than alternative solutions, such as smoothing or slowing down camera movements.

The contributions of this work include:

- perceptual studies of the relationship between simulator sickness and visual cues from virtual scenes and camera trajectory,
- a perceptual model for quantifying discomfort which is based on our perceptual studies,
- a real-time, online optimization for virtual camera control which reduces simulator sickness while maintaining original navigation designs.

2 RELATED WORK

2.1 Motion Perception and Simulator Sickness

Human brain infers motions of the body and surrounding objects based on visual, vestibular, and proprioceptive information. While the first one comes from observed images, the latter two depend solely on the physical motion of the subject. A human's motion perception from different motion cues are usually combined [DeAngelis and Angelaki 2011]. When the visually induced motion perception conflicts with physically induced motion perception, people may experience motion sickness [Keshavarz et al. 2015], such as carsickness and seasickness. Similarly, in VR, motion sickness can be induced when the movement of the user's avatar in virtual world is inconsistent with the user's physical movement in real world [McGill et al. 2017; Riecke et al. 2012; Xiao and Benko 2016]. According to one of the widely accepted explanations for the initiation of motion sickness, the sensory conflict theory [Reason and Brand 1975], this mismatch is especially agitating when translational acceleration or rotation occurs in the virtual world. Specifically, in human's vestibular system, each otolith organ acts primarily as an accelerometer measuring linear acceleration and each semicircular canal acts primarily as a gyroscope measuring angular velocity [Bos et al. 2008]. In other words, given negligible physical motion input, VVC mainly comes from the visual stimuli which induce illusory self-acceleration or self-rotation perception [Bos et al. 2008; LaViola Jr 2000; So et al. 2001a,b].

Both visually induced self-acceleration and self-rotation perception belong to the illusory self-motion created by the visual stimuli, also called vection [Ash et al. 2013; Dichgans and Brandt 1978]. Various potential factors affecting vection have been examined. The object and background hypothesis [Seno et al. 2009] suggests that the motion of the background induces vection while the motion of the objects (foreground) reduces the vection induced by the background because they generate an "inverted vection" [Nakamura and Shimojo 2000]. The hypothesis was further confirmed in [Seya et al. 2014]. The studies in [Nakamura et al. 2016] have also demonstrated that the relative motion between the object and the background plays an important role in inducing self-motion perception.

In agreement with the background hypothesis, several works examined the vection induced by objects at different depths [Brandt et al. 1975; Howard and Heckmann 1989; Nakamura 2006; Seya et al. 2014]. They show that distant objects are taken as background and induce direction-consistent vection, while foreground objects induce little vection. They further argue that the depth order is a more important factor than the absolute depths at which the objects are located. The consequence of these observations is the fact that the direction of self-motion can be often estimated using optical flow [Lappe et al. 1999], but in some cases, extra retinal signals are needed to disambiguate problematic situations [Lappe et al. 1999].

A more general perception of object motion is also affected by scene complexity and rendering [Allue et al. 2016], as well as by object-wise distances, size, and motion parallax [Distler et al. 2000]. Therefore, these factors also contribute to vection and VIMS. Motion perception also depends on the localization of the stimulus in the visual field. Peripheral stimuli have been shown to dominate motion perception [Brandt et al. 1973; Kim and Kim 2019; Previc and Donnelly 1993; Seno et al. 2009]. Kim and Kim [2019] specifically report the retinal periphery is more susceptible to VIMS in VR.

There have been many investigations reporting how vection and VIMS are related. Although the conclusion is still arguable, it is widely accepted that vection does not necessarily cause VIMS, and linear-acceleration-related vection and rotation-related vection can induce VIMS. Therefore, inspired by the above findings, we model the above factors affecting the vection of linear acceleration and the vection of rotation in our perceptual investigation. Following the finding of Lappe et al. [1999], we also make direct use of optical flow information which is available during rendering.

2.2 Reducing Visual-Vestibular Conflict

Simulator sickness is a widely acknowledged problem and a critical challenge in today's VR systems. An excellent overview of practical techniques to reduce the effect of the visual-vestibular conflict is provided in [Yao et al. 2014]. The most straightforward techniques try to avoid situations when the conflict can be triggered, for example, by avoiding accelerations or introducing different means of convening movement (e.g., teleportation). Other techniques focus on manipulating visual stimuli to limit vection. Examples include limiting the field of view [Fernandes and Feiner 2016] or blurring the image during rotational movement [Budhiraja et al. 2017]. All the above techniques have a significant drawback of modifying the user experience either by introducing substantial changes to the displayed content or restricting the actions performed by the user.

In contrast, the goal of our work is to maintain the initial design of the user experience as much as possible, while minimizing the effect of VVC. We share this goal with the work of Rietzler et al. [2018], which have proposed to stimulate user's vestibular system using rotation impulse in order to induce self-motion that can reduce VVC. Another related effort is the machine learning approach to estimate VR cybersickness [Padmanaban et al. 2018]. However, unlike our approach, their work does not provide a systematical solution for reducing simulator sickness.

2.3 Camera Path Optimization

A goal of this paper is to reduce simulator sickness through perception-aware camera path planning. Different planning approaches have been extensively proposed in interaction, animation and films production [Bell et al. 2001; Christie et al. 2008; He et al. 1996]. For example, Jardillier et al. [1998] have modeled vantage angle, ramming, and object sizes as constraints. Argelaguet et al. [2010] have adjusted the speed curve via optical flow and saliency map. Their perceptual metrics quantifies factors for visually-pleasant animations. Similarly, other research has been proposed to maximize scene coverage during a fly-through [Huang et al. 2016]. Those methods are, however, mainly for story-telling/entertainment and traditional display platforms.

Optimization-based methods outperform constraint-based ones with the help of flexible goal-driven modeling in different scenarios, such as VR locomotion [Sun et al. 2018]. Our idea is most similar to modeling the scene as a potential field. Hong et al. [1997] and Chiou et al. [1998] have modeled the destination and collision as attractive and repulsive fields, respectively. However, the local minimum may trap the search. To overcome this difficulty, [Burelli and Yannakakis 2010] have applied a stochastic population-based global search in the potential field. Another way to model the environment is to discretize the space into cells. For example, [Andújar et al. 2004] have presented a graph structure that embodies the cell connectivity. For faster performance, Christie et al. [2008; 2005] have proposed *semantic volumes* to reduce the searching space. [Lino and Christie 2015] have introduced the Toric space, a representation to reduce the search space from 7D to 3D.

To our best knowledge, no existing path planning approach aims at reducing VR sickness.

3 VISUAL-VESTIBULAR CONFLICT MODEL

In this section, we present a series of perceptual experiments to investigate scene and camera factors that can potentially affect VIMS. We then use the collected data to build a computational model of VIMS. Our studies simulate the common scenarios in which the user remains seated and explores the virtual environment by rotating her/his head and translating via hand-held controllers. Such stationary stances induces negligible vestibular motion sensation and therefore, VVC mainly comes from relative motion between the virtual scene and the camera (Section 2). This relative motion is perceived and interpreted by the human visual system via retinal optical flows [Warren 2004]. To model visual discomfort, we consider variables that can impact optical flows such as camera motions and object depths, and other perceptual factors such as eccentricity due to the motion sensitivity of peripheral vision (Section 2). Below, we present experiments analyzing the impact of depth, translational acceleration and rotational velocity on VIMS. The findings later motivate and inform our design of a new computational model for predicting VIMS G from a set of attributes $\mathbf{u}$ (Section 3.3):

$$G = \mathcal{L}(\mathbf{u}). \tag{1}$$

The goal of Section 3 is to understand *what* are the factors defining $\mathbf{u}$ and *how* they contribute to G.

3.1 Study - Scene Depth

Scene depth is a direct factor on the relative motion between objects and camera motion. Intuitively, scene/object depth should be negatively correlated with perceived VVC, since a closer object results in larger retinal optical flows. In addition to motion parallax depth cues, head-mounted displays also provide binocular disparity depth cues, which can contribute substantially to the perception of scene layout [Cutting and Vishton 1994] and compensate for the smaller retinal velocities of distant objects. In this experiment, we evaluate whether depth has indeed a significant impact on VVC in VR systems.

Stimuli. To test the influence of depth on VVC, we designed three scenes: Spheres, Forest, and Town (Figure 2), which provide visual stimuli with different environments, object depths, and field-of-view coverage. In particular, the Spheres scene provides simple visual stimuli requiring little cognitive understanding of the spatial structure of the environment, the Forest scene provides a natural landscape, while the Town scene provides a simple and yet familiar urban setting. For each scene, the virtual camera was placed in three different locations to simulate scene depths (3 m, 6 m and 9 m for Spheres; 50 m, 100 m and 150 m for Forest; 30 m, 45 m and 60 m for Town). The camera was moving horizontally with the same constant acceleration resulting in 60-seconds long stimuli. To ensure that depth was the only factor influencing perception, for each camera position the geometry was designed to maintain the same projected on-screen size.

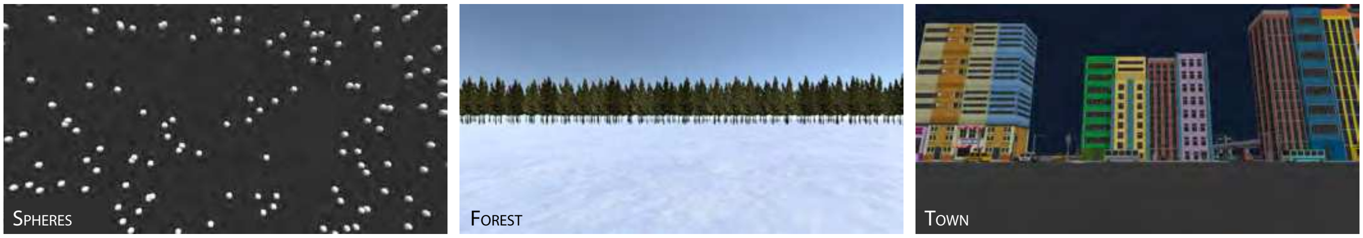


Fig. 2. *The scenes used in Section 3.1.*

Participants. Twelve subjects (5 female, 7 male) with ages 24-38 participated in the study. They had normal or corrected-to-normal vision. The participants were naïve to the hypothesis tested in the experiment.

Task. We used the *two-alternative forced choice* (2AFC) method, where each trial consisted of two animations shown sequentially from the same scene but at different distances using an HTC Vive headset. The order of the two animations was randomized for all trials across participants. The participants were instructed to indicate the animation causing stronger discomfort. To avoid discomfort accumulation there was a no less than 30-second break between the two animations and no less than 60-second break between consecutive trials. The participants were also instructed to take breaks or terminate the experiment early if they experience any signs of fatigue. To restrict the length of the experiment, each participant evaluated two comparisons of each scene, six comparisons in total. Participants were instructed to look straight ahead with no head rotation during each animation.

Results. 11 participants completed the experiment. One participant reported Trypophobia to dense spheres in the Spheres scene. Thus, his votings in this scene were dropped to avoid the medical condition reflected on VIMS. We analyzed the results of the experiment using binomial test which revealed a significant effect of depth on VIMS ($p < 0.001$). A post-hoc analysis showed that most participants indicated the animation with smaller camera distance as the one inducing stronger discomfort. Moreover, the effect size can be considered large since 84 % of answers agreed with this statement. Based on this study, we conclude that scene/object depth significantly contributes to VIMS in VR. The closer the visual stimuli are to the camera, the stronger the VIMS. Therefore, in Equation (1), $\mathbf{u}$ should include scene depth as an attribute.

Please note that the scenes in this experiment do not contain clear foreground/background separation. Also, the only variable differentiating the stimuli is the camera distance. As a result, our analysis neither investigates nor incorporates the background hypothesis [Seno et al. 2009], and models only the fact that the same object located at different distances contributes to VIMS differently. Consequently, our computational model presented in Section 3.3 does not make an explicit distinction between foreground and background.

3.2 Study - Camera Motion

In this subsection, we analyze how camera motion impacts VIMS. Since constant velocity usually does not introduce significant discomfort [Yao et al. 2014], we focus on VIMS induced by acceleration. Specifically, we study the camera acceleration along linear trajectory and camera rotation. Although these factors have been shown to impact motion sickness (Section 2), our goal is to quantify this impact.

Stimuli. We have designed a retail store scene (Figure 3a) containing multiple objects. The goal here was to create a scene from peoples' daily life, which encourages them to immerse themselves into a virtual environment. We prepared different viewing conditions. In the first scenario, the virtual camera was moving on a linear trajectory with five different translational acceleration values $\mathbf{T} \in \{1.5, 6.0, 24.0, 48.0, 75.0\, m/s^2\}$. In the second scenario, the virtual camera was rotating in yaw with five different rotational velocities $\mathbf{R} \in \{2.0, 8.0, 32.0, 64.0, 100.0\, ^\circ/s\}$. In addition, the scene with a static camera served as a minimal-discomfort stimuli. Each of the 11 different stimuli was 10 seconds long. To minimize discomfort introduced by vergence-accommodation conflicts, the scene content was designed to lie within the comfort range for our head-mounted display setup [Shibata et al. 2011].

Participants. 11 subjects (4 females and 7 males, with ages between 23 and 30) participated in this study. Similar to the previous experiment, they had a normal or corrected-to-normal vision and were naïve about the hypothesis tested in the experiment.

Task. Similar to the previous experiment, this study also followed a 2AFC design. In each trial, two stimuli were sequentially shown to participants. They were randomly chosen from the two scenarios described above. Each trial contained one animation with translational acceleration and one with constant rotational velocity. This allowed us to compare VIMS induced by translational acceleration and rotational velocity. The subjects were instructed to indicate one of the two stimuli with less severe discomfort. The stimuli were shown with a minimal 30-second in-between break, and there was a minimal 60-second break between trials. The participants were free to terminate the experiment early. To ensure proper focus and consistent experiences, the participants were instructed to keep their heads stationary (without rotation), fixate on a tumbling letter "E" positioned in the center of the screen and press the upper-arrow keyboard key when the letter was oriented upwards. The inter- and intra-participants sequences were randomized. Each participant conducted all comparisons once.

(a) Scene (b) Optical flow visualization

Fig. 3. *Stimuli used in Section 3.2.* (a) Shows the scene content. We used the letter "E" for gaze fixation. (b) Shows a sampled optical flow where the green arrows on the view indicate the magnitude while the camera rotates.

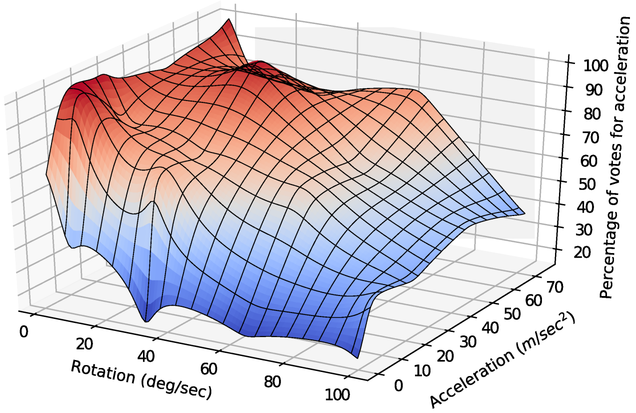


Fig. 4. *Camera motion study result.* Plotting user voting for linear acceleration versus rotation in Section 3.2. The height is a percentage number indicating the portion of votes for the acceleration stimuli.

Results. All participants completed this experiment. Figure 4 visualizes the results of the study. They show a positive correlation between translational acceleration/rotational velocity and VIMS. The participants reported that trials with translational acceleration $= 75.0m/s^2$ and rotational velocity $= 100°/s$ were too fast to observe noticeable sickness. We attribute this to the fact that beyond a certain velocity threshold it is difficult for subjects to track and interpret the scene. Consequently, we reject the results for these stimuli for further consideration. We analyze the experimental data using ANOVA. The results reveal that both translational acceleration ($F = 9.675, p = .004$) and rotational velocity ($F = 8.325, p = .006$) have significant effect on user discomfort votings. We will use the results obtained here to later train our computational model for predicting VIMS (Section 3.3).

3.3 The Analytical Model for VIMS

Based on prior perceptual studies and our own experiment results, we model VIMS as a function of translational acceleration, rotational velocity, and object depth. Since we consider complex scenes, we compute this information locally for individual pixels. For an object at pixel (i, j), we define $a_T(i, j)$ as the translational acceleration, $\omega_R(i, j)$ as the rotational velocity and $d(i, j)$ as the depth, all measured relative to the virtual camera.

Model components. We combine these quantities in a computational model of VIMS using two main terms:

$$I_T(i, j) = a_T(i, j) \cdot \Delta t^2 / d(i, j) \tag{2}$$

$$I_R(i, j) = \omega_R(i, j) \cdot \Delta t, \tag{3}$$

where $I_T(.)$ and $I_R(.)$ account for the simulator sickness induced by the relative acceleration and rotation. In both terms, the corresponding velocities are normalized by the frame duration Δt. Additionally, we account for the effect of depth in $I_T()$ and normalize the acceleration, a_T, by $d(i, j)$. We do not apply this normalization to the rotation component since, while rotating the camera, the movement of an object in the visual field does not depend on its depth.

Our model also accounts for the recent discovery that VIMS is dominated by peripheral vision [Kim and Kim 2019]. That is, the same motion observed at different locations in the visual field has different impacts on the VIMS. To model this, we weigh the contribution of the terms from Equations (2) and (3) based on the eccentricity.

Model. We define our predictor for VIMS as:

$$\mathcal{L} = \sum_{(i,j)}^{I_w, I_h} W(i, j) \cdot \left(w_T \cdot I_T(i, j) + w_R \cdot I_R(i, j)\right), \tag{4}$$

where I_w is image width, I_h is image height, w_T and w_R are parameters which weigh the translational and rotational terms of our model and will be calibrated using our experimental data.

$$w(i, j) = 1 - g\left((2i - I_w)/I_w, (2j - I_h)/I_h, \sigma\right), \tag{5}$$

$$W(i, j) = w(i, j) \Big/ \sum_{(i,j)}^{I_w, I_h} w(i, j), \tag{6}$$

where $w(i, j)$ is a per-pixel Gaussian-based weighting function which models larger influence of the peripheral vision on vection. Here, $g(x, y, \sigma)$ represents the value at point (x, y), which scales the value of a 2D Gaussian function (with standard deviation σ) to $[0, 1]$. Similar to other parameters of our model, we find the best σ value in our calibration step.

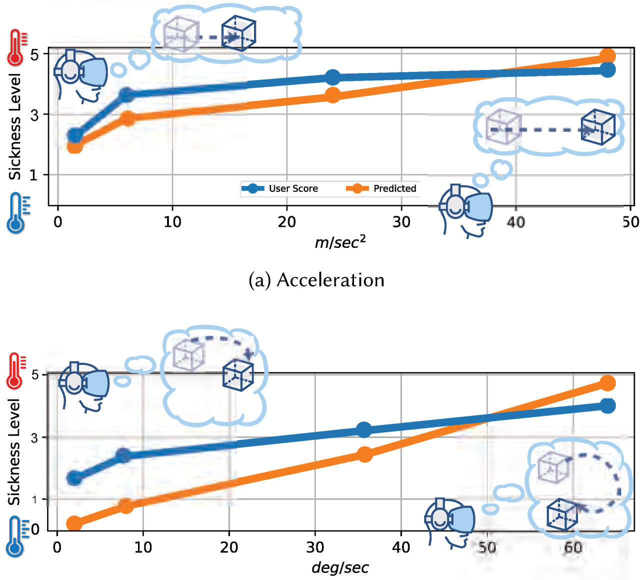
Model Calibration. Our perceptual data from Section 3.2 describes only a binary relation between the discomfort experienced for different stimuli. Since our goal is to devise a perceptual model which quantifies the amount of discomfort induced by a given stimulus, we adapt a Bayesian method [Kiran Adhikarla et al. 2017; Silverstein and Farrell 2001] to convert our pairwise comparison data into scalar data. The obtained scores provide a perceptual measure of the simulator sickness for each stimulus, which we use to train our model in Equation (4) to directly predict the scores. As the scores from the experiment correspond to the judgments made for entire videos, and our model provides a prediction for a single frame, the training has to be performed on accumulating scores. To this end, we assume that the scores from the experiments should correspond to the scores predicted by our model averaged across the duration of the whole animations. Consequently, for training, we define the prediction for the stimuli in our perceptual experiments as:

$$G(T) = \frac{1}{T} \sum_{i \in [0,T]} \mathcal{L}_i, \tag{7}$$

where $\mathcal{L}_i$ is a prediction calculated for frame i according to Equation (4); T denotes the duration (in terms of number of frames) of the sequence.

We train our model by minimizing the root mean square error between the prediction and the scores for the eight motions tested in our perceptual experiments. The obtained parameters for the model are: $w_T = 28.9$, $w_R = 28.4$, $\sigma = 1.99 I_h$. To test the ability of the model to handle new data, we also evaluate it using 4-fold cross-validation. Each fold contains 8 sets, including 6 for training and 2 for validating. This gives us $\binom{8}{2} = 28$ folds. The average error across all of them was 0.88 ($SD = 0.30$). Given the range of considered scores (Figure 5), we consider the error to be sufficiently low.

We discover that the trained large value of σ indicates gradually changing (instead of mostly uniform) weights along retinal eccentricity. This matches the experimental results from [Kim and Kim 2019].



(a) Acceleration

(b) Rotation

Fig. 5. *Comparing the approximated subject scores from Section 3.2 (in blue) with our fitted model from Section 3.3 (in orange).* A higher score indicates a stronger perceived discomfort.

4 OPTIMIZING CAMERA CONTROL

With the perceptual model in Equations (1) and (4), our goal is to minimize VIMS through automatic and progressive virtual camera planning in real-time. There are straightforward solutions, such as slowing down the camera movements to cause low values for $I_T()$ or $I_R()$. However, we should also respect the original intent of the content creators who design the camera movements for story-telling, cinematography, or gaming purposes. Consequently, we present a real-time camera motion optimization system which reduces VIMS while respecting the original movement.

4.1 Minimizing VVC Discomfort

We adapt the VIMS model from Equations (2) to (4) to optimize translational acceleration a_T and rotational velocity ω_R for reducing VIMS in the immediate next frame. We model the energy cost of VVC discomfort in our optimization as:

$$E_u(a_T, \omega_R) = \sum_{(i,j)}^{I_w, I_h} W(i,j) \left(w_T \|I_T(i,j)\|^2 + w_R \|I_R(i,j)\|^2 \right). \tag{8}$$

As described in Section 3.3, $I_T(i,j)$ and $I_R(i,j)$ are translational and rotational optical flows, and w_T, w_R are *time-independent* model parameters estimated over multiple frames via Equation (7). Given a VR scene, we need to make an estimation of each visible object's a_T and ω_R to analyze $I_T(i,j)$ and $I_R(i,j)$ introduced by visible objects in the environment.

Estimating a_T. Given frame k and a pixel (i,j) from object o, our system should plan the optimal camera motion for the next frame $k+1$, in which the relative (i.e., perceived) velocity between the virtual camera (with velocity $\mathbf{v}_{c,k+1}$) and the object (with velocity $\mathbf{v}_{o,k+1}$) is:

$$\begin{aligned} \mathbf{v}^c_{o,k+1} &= \mathbf{v}_{o,k+1} - \mathbf{v}_{c,k+1} \\ &\approx \mathbf{v}_{o,k} + \mathbf{a}_{o,k}\Delta t - \mathbf{v}_{c,k+1}, \end{aligned} \tag{9}$$

where $\mathbf{v}^c_{o,k+1}$ denotes the estimated $\mathbf{v}^c_o$ in frame $k+1$, $\mathbf{v}_{o,k}$ and $\mathbf{a}_{o,k}$ are the object's current velocity and acceleration, and the future camera velocity $\mathbf{v}_{c,k+1}$ can be estimated from the current camera velocity $\mathbf{v}_{c,k}$ and user controller inputs. Thus, the perceived acceleration can be obtained as:

$$\begin{aligned} a_T \Delta t &= \mathbf{v}^c_{o,k+1} - \mathbf{v}^c_{o,k} \\ &= \mathbf{v}_{o,k} + \mathbf{a}_{o,k}\Delta t - \mathbf{v}_{c,k+1} - \mathbf{v}^c_{o,k} \\ &= \mathbf{a}_{o,k}\Delta t - \left(\mathbf{v}_{c,k+1} - \mathbf{v}_{c,k}\right). \end{aligned} \tag{10}$$

Estimating ω_R. An object rotation in world space, $\mathbf{q}_o$ as represented by a quaternion, can be decomposed into a rotation around the camera ($\mathbf{q}^c_o$) and a translation along the axis connects the camera and the object ($\mathbf{t}^c_o$):

$$\mathbf{q}_o = \left[\mathbf{q}^c_o, \mathbf{t}^c_o\right]. \tag{11}$$

Given the camera's facing direction $\mathbf{n}_k$, assuming the object rotation relative to the camera is approximately similar between adjacent frames, the perceived object rotation velocity becomes:

$$\begin{aligned} \omega_R &= (\mathbf{q}^c_{o,k+1} - \mathbf{q}_{c,k+1})\mathbf{n}_k \\ &\approx (\mathbf{q}^c_{o,k} - \mathbf{q}_{c,k+1})\mathbf{n}_k, \end{aligned} \tag{12}$$

where quaternion $\mathbf{q}^c_{o,k+1}$ is the estimated $\mathbf{q}^c_o$ in frame $k+1$, and $\mathbf{q}_{c,k+1}$ represents the camera rotation from frame k to $k+1$. Equation (12) shows that ω_R can be controlled by $\mathbf{q}_{c,k+1}$.

The Cost. Based on the above analysis and discussion, at a given frame k our goal of minimizing VVC can be expressed as:

$$\underset{a_T,\,\omega_R}{\text{minimize}} \quad E_u\left(a_T, \omega_R\right) \tag{13}$$

which is approximately equivalent to:

$$\underset{\mathbf{v}_{c,k+1},\,\mathbf{q}_{c,k+1}}{\text{minimize}} \quad E_u\left(\mathbf{v}_{c,k+1}, \mathbf{q}_{c,k+1}\right) \simeq \sum_{(i,j)}^{I_w, I_h} W(i,j) \left(\begin{array}{c} w_T \left\|\left(\mathbf{v}_{c,k} + \mathbf{a}_{o,k}\Delta t - \mathbf{v}_{c,k+1}\right)\Delta t / d_o\right\|^2 \\ + w_R \left\|\left(\mathbf{q}^c_{o,k} - \mathbf{q}_{c,k+1}\right)\mathbf{n}_k \Delta t\right\|^2 \end{array} \right). \tag{14}$$

Therefore, finding the optimal $\mathbf{v}_c$ and $\mathbf{q}_c$ is the goal of minimizing VIMS in each frame. In our implementation, we accelerate the pixel-wise computation cost using a GPU, as detailed in Section 5.

4.2 Preserving Original Path

In a pre-scripted VR navigation, the designers can predefine the original camera movements. To observe the original path, we consider its parameters represented as camera position ($\hat{\mathbf{p}}$) and forward direction ($\hat{\tau}$). When designing an animation, creators typically specify target camera positions, a.k.a., points of interest (*PoI*), along the timeline. To enable numerical optimization (with C^1 continuity) between these *PoI*s, we represent the output path as a set of parameterized 3D Bézier curve $\mathbf{p}(\theta_k)$, where $0 \le \theta_k \le 1$ is the curve parameter at the k-th frame in each Bézier curve.

For active navigation (via user's self-control) without a given original path, we estimate it via existing motion prediction or extrapolation methods. Given a user's position $\mathbf{P}_k$ and velocity $\mathbf{v}_{c,k}$ in the current frame and the user's input of moving velocity $\mathbf{v}_{user}$ for the next frame, in our current implementation we estimate the user's active navigation path locally via Catmull-Rom spline with 4 control points:

$$\begin{aligned} C_0 &= \mathbf{P}_k - N\mathbf{v}_{c,k}\Delta t, \\ C_1 &= \mathbf{P}_k, \\ C_2 &= \mathbf{P}_k + N\mathbf{v}_{user}\Delta t, \\ C_3 &= \mathbf{P}_k + 2N\mathbf{v}_{user}\Delta t, \end{aligned} \tag{15}$$

where N is used to describe the number of frames with which we predict the user's movement. Empirically, we set $N = 20$ in our current implementation. Note that other motion prediction methods (e.g., Kalman filtering) can be orthogonally combined with our system. We chose Catmull-Rom spline with 4 control points as it works reasonably well and is simple to implement. We decompose the motion preserving goal as two subproblems in spatial and temporal domains.

Spatial Matching. The goal of this step is to match the overall camera positions between the original path and the VIMS-optimized path. Essentially, in this path following problem, given the original path $\hat{\mathbf{p}}$, this match is to determine the optimal $\mathbf{p}(\theta)$ on each frame by minimizing the contour error and lag error between $\hat{\mathbf{p}}$ and $\mathbf{p}(\theta)$ [Lam et al. 2010]. However, the contour error is not suitable to directly serve as an error measure in the path following approach, because it involves in an optimization problem over the entire path [Lam et al. 2010; Nägeli et al. 2017]. We adapt the contour error and lag error approximation of [Nägeli et al. 2017] in our spatial matching. We define the relative vector $\bar{\mathbf{p}}$ as the vector from the camera to the desired position on the path:

$$\begin{aligned} \bar{\mathbf{p}}(\mathbf{v}_{c,k+1}) &\triangleq \mathbf{p}(\theta_{k+1}) - \hat{\mathbf{p}}(k+1) \\ &= \mathbf{p}(\theta_k) + \Delta t\mathbf{v}_{c,k} - \hat{\mathbf{p}}(k+1). \end{aligned} \tag{16}$$

For every frame, the contour matching error is:

$$e_c(\mathbf{v}_{c,k+1}) = \|\bar{\mathbf{p}} - (\bar{\mathbf{p}} \cdot \hat{\tau}_{k+1})\hat{\tau}_{k+1}\|_2\,, \tag{17}$$

while the lag error is:

$$e_l(\mathbf{v}_{c,k+1}) = \|\bar{\mathbf{p}} \cdot \hat{\tau}_{k+1}\|_2\,. \tag{18}$$

Temporal Matching. The camera moves dynamically in typical VR navigation applications. That requires matching the original path in the temporal domain as well. The temporal matching is defined as

$$e_t(\mathbf{v}_{c,k+1}) = \left\|\theta_{k+1} - \hat{\mathbf{p}}^{-1}(k+1)\right\|_2\,, \tag{19}$$

where $^{-1}$ denotes the inverse mapping from a position $\mathbf{p}()$ on the Bézier curve to parameter θ. In practice, the content creator can specify *PoI*s similar to common animation/video editing software. Our system automatically matches their intention of story-telling or task design from these key frames.

Overall, the original path preserving formulas are combined as:

$$E_p(\mathbf{v}_{c,k+1}) = \alpha_l e_l^2 + \alpha_c e_c^2 + \alpha_t e_t^2, \tag{20}$$

where α_l, α_c and α_t are weights. In our evaluation, we set $\alpha_l = 5$, $\alpha_c = 0.3$ and $\alpha_t = 8$.

4.3 Optimization

We combine the cost energy of VVC discomfort in Equation (14) and path matching in Equation (20) to model the VR navigation optimization problem. In each frame, the camera position is determined by conducting the following optimization:

$$E(\mathbf{v}_{c,k+1}, \mathbf{q}_{c,k+1}) = E_u + \lambda E_p, \tag{21}$$

where λ is the weight of the cost term which can be tuned based on different navigation preferences (set as $\lambda = 150$ in our experiments). To stabilize the change of camera velocity on adjacent frames, we add a regularization term to our cost function:

$$\begin{aligned} \underset{\mathbf{v}_{c,k+1},\,\mathbf{q}_{c,k+1}}{\text{minimize}} \quad & E(\mathbf{v}_{c,k+1}, \mathbf{q}_{c,k+1}) + \lambda_0 \left\|\mathbf{v}_{c,k+1} - \mathbf{v}_{c,k}\right\|_2^2 \\ \text{subject to} \quad & 0 \le \theta_k \le 1 \end{aligned} \tag{22}$$

where λ_0 is the weight of the regularization term ($\lambda_0 = 0.003$ in our experiments).

5 IMPLEMENTATION

Optimization strategy. In Equation (22), at each frame k, we estimate the optimized camera position and orientation for the following N frames. Only the camera position and orientation for frame $k+1$ is applied in the camera control. This idea is from the optimization strategy in Model Prediction Control [Lam et al. 2010] for dynamic smooth control:

$$\begin{aligned} \underset{\mathbf{v}_{c,k+i},\,\mathbf{q}_{c,k+i}}{\text{minimize}} \quad & \sum_{i=1}^{N} E(\mathbf{v}_{c,k+i}, \mathbf{q}_{c,k+i}) + \lambda_0 \left\|\mathbf{v}_{c,k+1} - \mathbf{v}_{c,k+i-1}\right\|_2^2 \\ \text{subject to} \quad & 0 \le \theta_k \le 1 \end{aligned} \tag{23}$$

We used $N = 20$ in our system.

Object-wise acceleration. In Equation (8), the sum through all pixels (i, j) is atomic, thus computationally heavy. Alternatively, we take advantage of the compute buffer in the GPU to accelerate the projected area computation. For each frame k, each object's translational acceleration $\mathbf{a}_{o,k}$, rotation $\mathbf{q}_{o,k}$, camera facing direction $\mathbf{n}_k$, camera up direction, and camera position are sent to the rendering pass. Using this information, each pixel's $w^c_{o,k}$ are computed in the object rendering pass. $\mathbf{a}_{o,k}$ and $w^c_{o,k}$ are recorded in each pixel in a read/write buffer while rendering the objects. At the end of each scene rendering, this read/write buffer is sent to a compute shader to compute the sum of weighted translational acceleration as well as rotation velocity relative to the camera. The summation is implemented in the compute shader using $InterlockedAdd()$ function.

Hardware. The system is driven by a PC with Intel i9-7900X CPU @ 3.30 GHz, 64GB RAM, and one NVIDIA GTX 1080 Ti graphics card. The VR navigations are shown to the users through a HTC Vive headset.

Software. Our rendering system is implemented in Unity Engine®. The camera control optimization in Section 4 is implemented using the optimization library FORCES Pro [Domahidi and Jerez 2014]. The C code is generated in Matlab and deployed in Visual Studio 2013. The data communication between Unity and Visual Studio is developed using an asynchronous messaging library, ZeroMQ [Hintjens 2013].

6 EVALUATION

We evaluate the performance of our technique with a series of user studies using passive and active path control. The experiments compare our method with simpler alternatives in terms of induced discomfort using subjective feedback and the simulator sickness questionnaire. Additionally, we analyze the deviation of the optimized paths from the original ones.

6.1 Passive Navigation

In this experiment, we evaluate the performance of our method in a scenario where viewers control camera orientation (via head rotational tracking) but not motion trajectory.

Stimuli. We designed two scenes (S_1 and S_2 in Figure 6) for this experiment. The outer space scene (S_1) simulates a static scene while the racing car scene (S_2) simulates a dynamic scene. We compared four camera trajectories (P, O, U, T) for each scene. The first is the original, manually predefined trajectory (P), that is path and speed, which presents the intent of the creator. The second (O) is the trajectory resulting from our automatic camera control applied to P to minimize VIMS. We also consider simpler alternatives which can potentially reduce VIMS. The first one (U) is the trajectory that follows the predefined path P but with uniform speed along the entire path. The other alternative solution (acceleration-thresholding navigation, T) maintains the intention of P but minimizes the acceleration. U, T and P have the same animation duration over the entire path. The duration of the stimuli was 27 seconds for S_1 and 55 seconds for S_2. Please see the supplemental video for a preview of the stimuli.

Participants. 16 subjects (6 females, mean age 28.3) participated in S_1 and S_2. They were naïve to the hypothesis of the experiment.

Task. During the experiment, the participants wearing HTC Vive headset explored our stimuli. They remained seated but were free to rotate their heads and look around. In each trial, participants were shown one pair of stimuli sequentially in random order. Each pair contains our solution (O) and one of the three alternatives (P, U, or T). The experiment followed a 2AFC procedure, and the subjects were asked to choose the sequence in which they felt more comfortable. After each navigation condition, the subjects were instructed to fill out the Simulator Sickness Questionnaire (SSQ) [Kennedy et al. 1993]. To minimize the effect of accumulating discomfort during the experiment, we enforced at least a 60-second break between the two conditions in each trial, but the participants were instructed to take as much time as needed to recover. During the experiment, there were several cases where the participants took longer breaks (up to five minutes in extreme cases), but nobody terminated the test because of the severe discomfort.

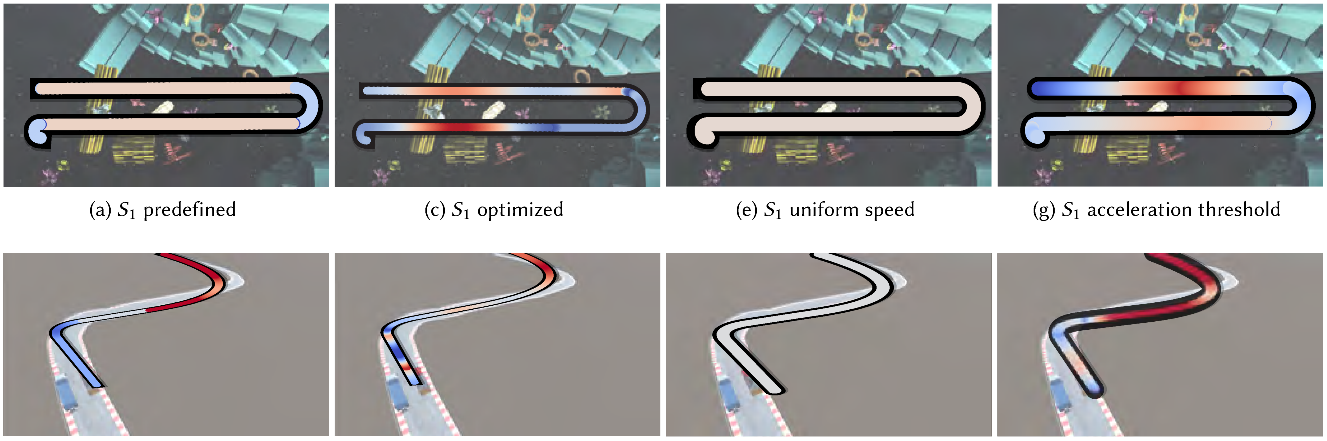
Results. The user preference votes are visualized in Figure 7. In all cases, most subjects rated our optimized condition (O) as being more comfortable than all alternative solutions. Binomial tests showed that differences between conditions O and P as well as O and T are statistically significant (p=.0013). Despite the difference in votes, the result was not statistically significant for the comparison between O and U.

The results of the SSQ were interpreted as nausea, oculomotor, disorientation, and total severity (TS) (Table 1) following [Kennedy et al. 1993]. We analyzed the data using ANOVA test to investigate the significance of the differences in TS scores between our method and the alternative solutions. The analysis showed that the difference between our method (O) and the predefined trajectory (P) as well as the difference between ours (O) and the acceleration threshold (T) are significant with $p = .041$ and $p = .004$, respectively. The difference between the strategy O and U was not statistically significant.

The above results demonstrate that our method can successfully optimize a predefined trajectory, and outperforms the acceleration threshold alternative. Even though the difference between our technique and the uniform speed alternative was not statistically significant, we will demonstrate later in Section 6.3 that our technique better follows the predefined trajectory.

6.2 Active Navigation

In the second experiment, users are allowed to actively control the navigation in addition to head rotation. This is to simulate real-world interactive scenarios, such as gaming. The magnitude of



(a) S_1 predefined (c) S_1 optimized (e) S_1 uniform speed (g) S_1 acceleration threshold

(b) S_2 predefined (d) S_2 optimized (f) S_2 uniform speed (h) S_2 acceleration threshold

Fig. 6. *Visualization of scenes and camera trajectories from the passive navigation experiment (Section 6.1).* The first row shows the four navigation conditions in S_1. The second row shows the four navigation conditions in S_2. The real-time camera control responds to users actions and scene content in each trial. Hence, the illustrated optimized camera movements (second column) are the samples in S_1 and S_2, respectively. Color warmth represents speed along the camera paths. Please refer to our video for detailed active navigation comparison.

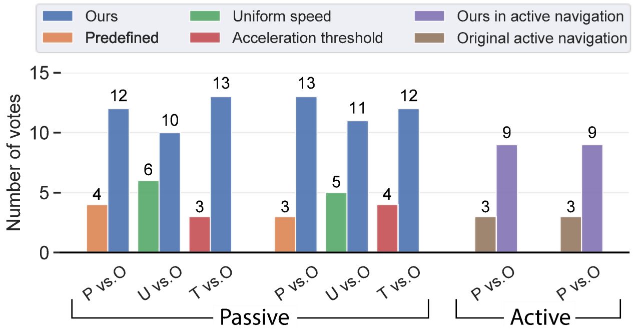


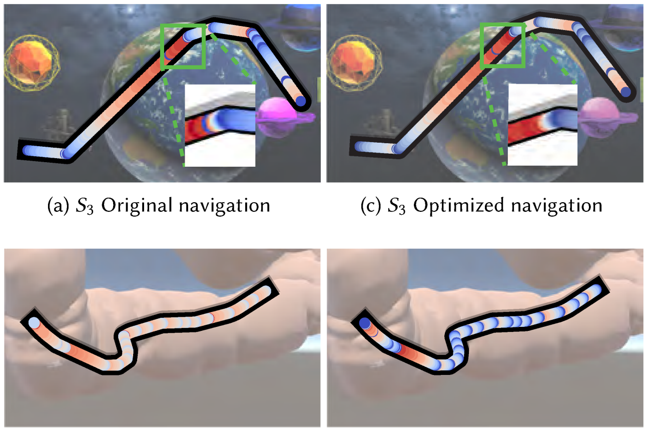
Fig. 7. *The users' preference votes from our evaluation experiment for passive and active navigation.* Higher numbers indicate preference towards the corresponding condition. For each navigation type, we provide two sets of plots which correspond to scene S_1, scene S_2, scene S_3, and scene S_4, respectively.

Table 1. *The averages and standard deviations (avg./SD) of SSQ scores obtained in the passive navigation experiment (Section 6.1).*

Condition	Nausea	Oculomotor	Disorientation	Total Severity
P	8.3/8.5	11/16	7.4/11	6.7/8.5
O	3.4/5.1	5.4/8.5	13/20	2.8/4.6
U	3.9/5.8	6.6/9.8	5.7/12	3.7/6.4
T	6.9/8.3	8.1/8.4	6.5/13	3.9/4.7

translation/rotation velocity is determined by the distance between the touchpad center and the pressed position.

Stimuli. We designed two new scenes for this experiment (S_3 and S_4 in Figure 8). The first is a flight simulator (S_3) with the earth providing strong visual motion cues. The second (S_4) is a scene simulating a virtual colonoscopy investigation via a confined and occluded environment. With these scenes, we tested two conditions: one where the camera movement is in full control of the user (P), and the second (O) where we apply our technique to minimize VIMS. We do not consider other simpler techniques used in our first experiment (Section 6.1), since the full trajectory is unknown in advance and computation of the average speed and the acceleration threshold is impossible.



(a) S_3 Original navigation (c) S_3 Optimized navigation

(b) S_4 Original navigation (d) S_4 Optimized navigation

Fig. 8. *Visualization of scenes and camera trajectories from the active navigation experiment (Section 6.2).* The first column shows original condition while the second column shows the corresponding optimized conditions from sampled trials. The real-time camera control responds to users' actions and scene content in each trial. Color warmth represents speed along the camera paths. The insets in S_3 shows local area zooming view. The major optimization reaction corresponds to users' forward moving operation. Please refer to our video for detailed active navigation comparison.

Participants. 12 subjects (4 females, mean age 26) participated in the experiment. All subjects were naïve to our hypothesis.

Task. The experiment followed a similar procedure to the one in Section 6.1. The main difference was that the participants were allowed to explore the environment by manipulating two HTC Vive controllers. One touchpad was used to change the velocity of the camera, while the second one was used to alter the forward direction. Before the experiment, the participants took part in a practice session to familiarize themselves with the setup. During the experiment, the subjects were asked to explore the virtual world using two different navigation methods (P and O) sequentially in random order. Each exploration was limited to five minutes, but participants were free to terminate at any time. They were encouraged to explore the virtual world as much as possible. Similar to the previous experiment, the participants were instructed to take a break between consecutive stimuli. At the end of the experiment, they were asked to indicate the more comfortable condition.

Result. Subjects' preference votes and SSQ scores are shown in Figure 7 and Table 2, respectively. The subjects voted the optimized condition (O) as being more comfortable than the original one (P). A binomial test showed that the difference is statistically significant with $p = .012$. An ANOVA test also showed a significant difference in the SSQ TS scores between the two techniques with $p = .032$. The analyses of both subjective preference votes and detailed SSQ scores suggest that our system can significantly reduce VIMS in active navigations.

Table 2. *The averages and standard deviations (avg./SD) of SSQ scores obtained in the active navigation experiment (Section 6.2).*

Condition	Nausea	Oculomotor	Disorientation	Total Severity
P	31.9/43.4	23.4/33.3	15.5/20.1	13.3/17.3
O	19.1/31.3	15.2/24.9	11.1/16.0	8.3/12.8

6.3 Intention Matching

As discussed in Section 4.2, the goal of our system is not only to reduce discomfort but also to preserve the creators' content intention expressed in the camera motion. Here, we propose a numerical measure to validate the effectiveness of reaching this goal.

Measurement. The goal of intention matching is to make the optimized camera motion close to the original one. Since the camera trajectory is defined using *PoI*s, between which the trajectory is interpolated, we evaluate the similarity between the trajectories by comparing the times when the corresponding *PoI*s are reached. Note that each of the considered solutions in our evaluation reaches all the *PoI*s. Consequently, we define our measure for the i-th *PoI* as:

$$e_m(i) = \left\|\hat{\mathbf{t}}_i - \mathbf{t}_i\right\|, \tag{24}$$

where $\mathbf{t}$ and $\hat{\mathbf{t}}$ are the times when the *PoI* is reached in the optimized and the predefined path, respectively.

Results. We compare e_m values derived from all 16 users' logs from the first experiment in Section 6.1 to compare our method to the uniform speed alternative. In both scenes, three *PoI*s specify the input trajectory. The averaged e_m values of each *PoI* in S_1 across all users are 0.04, 0.03, 0.06 seconds for our method, and 0.62, 0.27, 0.37 seconds for the uniform speed navigation. Similarly in S_2, the averaged e_m are 1.10, 1.49, 1.51 seconds for our method comparing to 6.14, 11.54, 11.63 seconds for the uniform speed navigation. Figure 9 depicts the results. The results for both scenes indicate that our technique achieves a closer match to the original path.

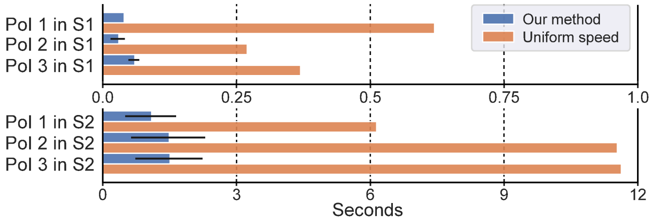


Fig. 9. *Matching error analysis.* The averaged motion matching error e_m between our method and the uniform speed navigation. The error bars indicate the standard deviation across the participants.

6.4 Summary

The combined results from Sections 6.1 and 6.2 show that our camera movement optimization improves perceived comfort over the original condition and the acceleration threshold alternative, while achieving comparable results to the uniform speed navigation. In Section 6.3, however, we demonstrate that our technique performs better in terms of matching the creators' original intention when compared to the uniform speed navigation. This joint analysis suggests that our method outperforms all tested alternative solutions when both comfort and intention matching are taken into account.

7 LIMITATIONS AND FUTURE WORK

Our perceptual model considered three main factors: scene depth, camera motion, and visual eccentricity. However, there are potentially other lower-level scene attributes that can contribute to VIMS induction [So et al. 2001a], such as colors, shapes, and textures, as well as individual perceptual factors influence different users' VIMS perception [Gibson 1950]. In addition, semantic factors can also influence user perception of sickness in VR exploration. It can be an interesting topic to further investigate how scene types influence VIMS perception, although we did not observe significant effect from our ANOVA analysis in Section 6. Peripheral vision is reported as dominating exocentric motion perception [Brandt et al. 1973] and contributing more to VIMS [Kim and Kim 2019]. However, the quantitative model is not established. Our perceptual model reflects this eccentric motion sensitivity via the inverse Gaussian-based weighting. We have not conducted a perceptual experiment to validate the design choice of using this particular weighting. The weighting model can be refined with further visual experiments. Our paper can serve as basis for more complete future models of VIMS.

Our camera path optimization can be applied to a full path, either known in advance or predicted on the fly. In our current study, we adopt a simple extrapolation scheme (Equation (15)) for active navigation. Predicting future user actions with machine learning or brain-computer interfaces may improve the optimization speed and further reduce discomfort. Beyond optimizing the camera control, adaptively manipulating virtual content, such as character motion or task object distribution, can be another interesting direction to explore.

To minimize VIMS induced by the camera-object relative motion, our method accelerates/rotates more slowly/gradually than the original in a static scene (e.g., the colonoscopy in Section 6.2 or animated example in the supplemental video). Since our optimization introduces local changes to the path, the resulting camera motion may look unnatural in some cases. For instance, when scene objects move with very high acceleration, our method may accelerate the camera to minimize the camera-object relative acceleration. This might cause significant decelerations in other periods during the navigation which may be perceive as unnatural motion. To prevent this, a designer can adjust λ. Alternatively, it might be possible to update the parameter dynamically.

Despite the complexity of our model, our solution does not receive significantly higher user preference when compared to the uniform speed solution in terms of reducing VIMS. It has to be noted, however, that our technique with the same level of VIMS is able to achieve significantly better intention matching.

In terms of the method implementation, the current system requires not only original camera motions but also scene object motions as inputs. This may lead to extra implementation complexity than the alternatives. However, we believe that the potential benefits given by such a scene analysis outweigh the additional requirements.

In our work, we assume that the viewer remains stationary, for example, seated in a chair. In the future, accounting for the limited, but possible, motion in the physical space would extend the applications of our technique. In particular, it would be interesting to combine our online camera path optimization with redirected walking techniques, such as [Dong et al. 2017; Sun et al. 2018].

8 CONCLUSIONS

While VR head-mounted displays are gaining more attention, the problem of discomfort induced by such devices starts affecting a larger number of users. One of the most significant sources of discomfort is simulator sickness. In this work, we present a real-time, content-aware, and automatic camera control approach, which can reduce the problem while maintaining the original intent regarding the camera path. The method is based on a series of perceptual experiments which allow us to derive a perceptual model for quantifying visually induced motion sickness. We believe that our studies and methods can also be applied to immersive film production, game design, and 360 video replay. In those scenarios, the original content creators, such as directors, designers, or regular customers, may not be aware of the potential discomfort when their materials are played in VR platforms. Our method can reduce simulator sickness and preserve the storytelling motion features encoded in the original camera path.

ACKNOWLEDGMENTS

We would like to thank Xiaojun Bi and Suwen Zhu for experiment analysis discussion, and the anonymous reviewers for their valuable suggestions. This project is partially supported by National Science Foundation grants NRT1633299, CNS1650499, and a gift from Adobe.

REFERENCES

Marcos Allue, Ana Serrano, Manuel G Bedia, and Belen Masia. 2016. Crossmodal Perception in Immersive Environments. In *CEIG '16*. 1–7.

Carlos Andújar, P Vázquez, and Marta Fairén. 2004. Way-finder: guided tours through complex walkthrough models. In *Computer Graphics Forum*, Vol. 23. 499–508.

Ferran Argelaguet and Carlos Andujar. 2010. Automatic Speed Graph Generation for Predefined Camera Paths. In *SG '10*. 115–126.

April Ash, Stephen Palmisano, Deborah Apthorp, and Robert S Allison. 2013. Vection in Depth During Treadmill Walking. *Perception* 42, 5 (2013), 562–576.

Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View Management for Virtual and Augmented Reality. In *UIST '01*. 101–110.

Jelte E Bos, Willem Bles, and Eric L Groen. 2008. A Theory on Visually Induced Motion Sickness. *Displays* 29, 2 (2008), 47–57.

Thomas Brandt, Johannes Dichgans, and E Koenig. 1973. Differential Effects of Central versus Peripheral Vision on Egocentric and Exocentric Motion Perception. *Experimental Brain Research* 16, 5 (1973), 476–491.

Thomas Brandt, Eugene R Wist, and Johannes Dichgans. 1975. Foreground and Background in Dynamic Spatial Orientation. *Perception & Psychophysics* 17, 5 (1975), 497–503.

Pulkit Budhiraja, Mark Roman Miller, Abhishek K Modi, and David Forsyth. 2017. Rotation Blurring: Use of Artificial Blurring to Reduce Cybersickness in Virtual Reality First Person Shooters. *arXiv preprint arXiv:1710.02599* (2017).

Paolo Burelli, Luca Di Gaspero, Andrea Ermetici, and Roberto Ranon. 2008. Virtual Camera Composition with Particle Swarm Optimization. In *International Symposium on Smart Graphics*. 130–141.

Paolo. Burelli and Georgios. N. Yannakakis. 2010. Global Search for Occlusion Minimisation in Virtual Camera Control. In *IEEE Congress on Evolutionary Computation*. 1–8.

Rui CH Chiou, Arie E Kaufman, Zhengrong Liang, Lichan Hong, and Miranda Achniotou. 1998. Interactive Path Planning for Virtual Endoscopy [Colon CT]. In *IEEE Nuclear Science Symposium. Conference Record.*, Vol. 3. IEEE, 2069–2072.

Marc Christie and Jean-Marie Normand. 2005. A Semantic Space Partitioning Approach to Virtual Camera Composition. In *Computer Graphics Forum*, Vol. 24. 247–256.

Marc Christie, Patrick Olivier, and Jean-Marie Normand. 2008. Camera Control in Computer Graphics. *Computer Graphics Forum* 27, 8 (2008), 2197–2218.

James Cutting and Peter Vishton. 1994. Perceiving Layout and Knowing Distances.

Gregory C DeAngelis and Dora E Angelaki. 2011. The Neural Bases of Multisensory Processes. Chapter Visual-vestibular integration for self-motion perception.

Johannes Dichgans and Thomas Brandt. 1978. Visual-Vestibular Interaction: Effects on self-motion perception and postural control. In *Perception*. 755–804.

Hartwig K Distler, Karl R Gegenfurtner, Hendrik AHC Van Veen, and Michael J Hawken. 2000. Velocity Constancy in a Virtual Reality Environment. *Perception* 29, 12 (2000), 1423–1435.

Alexander Domahidi and Juan Jerez. 2014. FORCES Professional. embotech GmbH (http://embotech.com/FORCES-Pro).

Zhi-Chao Dong, Xiao-Ming Fu, Chi Zhang, Kang Wu, and Ligang Liu. 2017. Smooth Assembled Mappings for Large-scale Real Walking. *ACM Transactions on Graphics* 36, 6, Article 211 (2017), 13 pages.

Ajoy S Fernandes and Steven K Feiner. 2016. Combating VR Sickness Through Subtle Dynamic Field-of-View Modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 201–210.

James J Gibson. 1950. *The Perception of the Visual World.* Oxford, England. xii+242 pages.

Li-wei He, Michael F. Cohen, and David H. Salesin. 1996. The Virtual Cinematographer: A Paradigm for Automatic Real-time Camera Control and Directing. In *SIGGRAPH '96*. 217–224.

Pieter Hintjens. 2013. *ZeroMQ: messaging for many applications.*

Lichan Hong, Shigeru Muraki, Arie Kaufman, Dirk Bartz, and Taosong He. 1997. Virtual Voyage: Interactive Navigation in the Human Colon. In *SIGGRAPH '97*. 27–34.

Ian P Howard and Thomas Heckmann. 1989. Circular Vection as a Function of the Relative Sizes, Distances, and Positions of Two Competing Visual Displays. *Perception* 18, 5 (1989), 657–665.

Hui Huang, Dani Lischinski, Zhuming Hao, Minglun Gong, Marc Christie, and Daniel Cohen-Or. 2016. Trip Synopsis: 60km in 60sec. *Computer Graphics Forum* 35, 7 (2016), 107–116.

Frank Jardillier and Eric Languénou. 1998. Screen-Space Constraints for Camera Movements: the Virtual Cameraman. In *Computer Graphics Forum*, Vol. 17. 175–186.

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

Behrang Keshavarz, Bernhard E Riecke, Lawrence J Hettinger, and Jennifer L Campos. 2015. Vection and Visually Induced Motion Sickness: how are they related? *Frontiers in Psychology* 6 (2015).

Nam-Gyoon Kim and Beom-Su Kim. 2019. The Effect of Retinal Eccentricity on Visually Induced Motion Sickness and Postural Control. *Applied Sciences* 9, 9 (2019), 1919.

Vamsi Kiran Adhikarla, Marek Vinkler, Denis Sumin, Rafal K Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. 2017. Towards a Quality Metric for Dense Light Fields. In *CVPR '17*. 58–67.

Denise Lam, Chris Manzie, and Malcolm Good. 2010. Model Predictive Contouring Control. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 6137–6142.

Eike Langbehn, Frank Steinicke, Markus Lappe, Gregory F Welch, and Gerd Bruder. 2018. In the Blink of an Eye: Leveraging Blink-Induced Suppression for Imperceptible Position and Orientation Redirection in Virtual Reality. *ACM Transactions on Graphics* 37, 4, Article 66 (2018).

Markus Lappe, Frank Bremmer, and AV Van den Berg. 1999. Perception of Self-Motion from Visual Flow. *Trends in Cognitive Sciences* 3, 9 (1999), 329–336.

Joseph J LaViola Jr. 2000. A Discussion of Cybersickness in Virtual Environments. *ACM SIGCHI Bulletin* 32, 1 (2000), 47–56.

Christophe Lino and Marc Christie. 2015. Intuitive and Efficient Camera Control with the Toric Space. *ACM Transactions on Graphics* 34, 4, Article 82 (2015), 12 pages.

Belen Masia, Gordon Wetzstein, Piotr Didyk, and Diego Gutierrez. 2013. Special Section on Advanced Displays: A Survey on Computational Displays: Pushing the Boundaries of Optics, Computation, and Perception. *Computers and Graphics* 37, 8 (2013), 1012–1038.

Michael E McCauley and Thomas J Sharkey. 1992. Cybersickness: Perception of Self-motion in Virtual Environments. *Presence: Teleoperators & Virtual Environments* 1, 3 (1992), 311–318.

Mark McGill, Alexander Ng, and Stephen Brewster. 2017. I Am The Passenger: How Visual Motion Cues Can Influence Sickness For In-Car VR. In *CHI '17*. 5655–5668.

Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. 2017. Real-time Planning for Automated Multi-view Drone Cinematography. *ACM Transactions on Graphics* 36, 4, Article 132 (2017), 10 pages.

Shinji Nakamura. 2006. Depth Separation between Foreground and Background on Visually Induced Perception of Self-Motion. *Perceptual and Motor Skills* 102, 3 (2006), 871–877.

Shinji Nakamura, Stephen Palmisano, and Juno Kim. 2016. Relative Visual Oscillation Can Facilitate Visually Induced Self-Motion Perception. *i-Perception* 7, 4 (2016), 1–18.

Shinji Nakamura and Shinsuke Shimojo. 2000. A Slowly Moving Foreground can Capture an Observer's Self-Motion - a Report of a New Motion Illusion: Inverted vection. *Vision Research* 40, 21 (2000), 2915–2923.

Nitish Padmanaban, Timon Ruban, Vincent Sitzmann, Anthony M Norcia, and Gordon Wetzstein. 2018. Towards a Machine-Learning Approach for Sickness Prediction in 360° Stereoscopic Videos. *IEEE Transactions on Visualization and Computer Graphics* 1 (2018), 1594–1603.

Fred H Previc and Michael Donnelly. 1993. The Effects of Visual Depth and Eccentricity on Manual Bias, Induced Motion, and Vection. *Perception* 22, 8 (1993), 929–945.

Jerrold D Prothero, Mark H Draper, DE Parker, and MJ Wells. 1999. The use of an independent visual background to reduce simulator side-effects. *Aviation, Space, and Environmental Medicine* 70, 3 Pt 1 (1999), 277–283.

Sharif Razzaque, David Swapp, Mel Slater, Mary C. Whitton, and Anthony Steed. 2002. Redirected Walking in Place. In *EGVE '02*. 123–130.

James T Reason and Joseph John Brand. 1975. *Motion sickness.*

Lisa Rebenitsch and Charles Owen. 2016. Review on Cybersickness in Applications and Visual Displays. *Virtual Reality* 20, 2 (2016), 101–125.

Bernhard E Riecke, Daniel Feuereissen, John J Rieser, and Timothy P McNamara. 2012. Self-Motion Illusions (Vection) in VR: Are they good for anything?. In *2012 IEEE Virtual Reality Workshops (VRW)*. 35–38.

Michael Rietzler, Teresa Hirzle, Jan Gugenheimer, Julian Frommel, Thomas Dreja, and Enrico Rukzio. 2018. VRSpinning: Exploring the Design Space of a 1D Rotation Platform to Increase the Perception of Self-Motion in VR. In *Proceedings of the 2018 Designing Interactive Systems Conference*. ACM, 99–108.

Takeharu Seno, Hiroyuki Ito, and Shoji Sunaga. 2009. The Object and Background Hypothesis for Vection. *Vision Research* 49, 24 (2009), 2973–2982.

Yasuhiro Seya, Takayuki Tsuji, and Hiroyuki Shinoda. 2014. Effect of Depth Order on Linear Vection with Optical Flows. *i-Perception* 5, 7 (2014), 630–640.

Takashi Shibata, Joohwan Kim, David M Hoffman, and Martin S Banks. 2011. The Zone of Comfort: Predicting Visual Discomfort with Stereo Displays. *Journal of Vision* 11, 8 (2011), 11–11.

D Amnon Silverstein and Joyce E Farrell. 2001. Efficient Method for Paired Comparison. *Journal of Electronic Imaging* 10, 2 (2001), 394–398.

Richard HY So, Andy Ho, and WT Lo. 2001a. A Metric to Quantify Virtual Scene Movement for the Study of Cybersickness: Definition, implementation, and verification. *Presence: Teleoperators and Virtual Environments* 10, 2 (2001), 193–215.

Richard HY So, WT Lo, and Andy TK Ho. 2001b. Effects of Navigation Speed on Motion Sickness Caused by an Immersive Virtual Environment. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 43, 3 (2001), 452–461.

Qi Sun, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan McGuire, David Luebke, and Arie E Kaufman. 2018. Towards Virtual Reality Infinite Walking: Dynamic Saccadic Redirection. *ACM Transactions on Graphics* 37, 4, Article 67 (2018), 13 pages.

Qi Sun, Li-Yi Wei, and Arie E Kaufman. 2016. Mapping Virtual and Physical Reality. *ACM Transactions on Graphics* 35, 4, Article 64 (2016), 12 pages.

William H. Warren. 2004. Optic Flow. In *The Visual Neurosciences*, Leo M. Chalupa John Simon Werner (Ed.). Cambridge, Chapter 10, 1247–1259.

Martin. Weier, Michael. Stengel, Thorsten. Roth, Piotr. Didyk, Elmar. Eisemann, Martin. Eisemann, Steve. Grogorick, Andr'e. Hinkenjann, Ernst. Kruijff, Marcus. Magnor, Karol. Myszkowski, and Philipp. Slusallek. 2017. Perception-driven Accelerated Rendering. *Computer Graphics Forum* 36, 2 (2017), 611–643.

Robert Xiao and Hrvoje Benko. 2016. Augmenting the Field-of-View of Head-Mounted Displays with Sparse Peripheral Displays. In *CHI '16*. 1221–1232.

Richard Yao, Tom Heath, Aaron Davies, Tom Forsyth, Nate Mitchell, and Perry Hoberman. 2014. Oculus VR best practices guide. https://scontent.oculuscdn.com/v/t64.5771-25/12482206_237917063479780_486464407014998016_n.pdf.