Event-Driven Lighting for Immersive Attention Guidance

Category: Research

Figure 1: todo

ABSTRACT

We present an framework for guiding users' attention in virtual environments by automatically detecting events of interest and adjusting the environment lighting to highlight these events. In immersive virtual environments, the user is free to view their virtual surroundings in any direction. For virtual experiences that are driven by a narrative, a common problem that occurs is that users may miss out on important events of the story line because the events took place outside of the user's field of view. To mitigate this problem, we developed a method to guide users' attention towards important events without breaking their immersion through nondiegetic cues like notification arrows. Our method accepts as input a 3D virtual scene representation and automatically detects events based on multimodal information (e.g., a narrative script, audio events, or collision events). We represent events as intervals of time with a name and a "payload", which may be empty or contain arbitrary data. Once an event is detected, our method adjusts the lighting in the scene to make the detected events more conspicuous to the user. To validate the efficacy of our method, we conducted a user study in which participants were tasked with learning the content of a virtual environment. Results of our study show that our system is able to significantly improve the user's ability to learn about and remember the layout of virtual environments. We conclude with additional examples that showcase the different situations in which our framework can be applied, including creating cinematic videos and directing users in interactive games.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 INTRODUCTION

In immersive virtual environments (VEs), the user is usually free to move the camera in any direction they please. This leads to a common problem in virtual reality (VR) experiences, wherein the user may turn to face away from an object or event of interest that is crucial to their virtual experience []. That is, the important events of the virtual experience occur outside of the user's field of view and they may become confused due to missing important information. To overcome this problem, researchers have developed different attention-guidance techniques [24]. Though their implementations differ, all attention-guidance techniques introduce some stimulus (usually visual) that attracts the user's gaze towards the virtual area of interest. When choosing an attention-guiding stimulus, it is important to manage the tradeoff between the stimulus' power (ability to reliably influence the user's attention) and how immersive the stimulus is.

Main Contributions: In this work, we introduce an *event-driven lighting* framework for attention-guidance in immersive virtual environments. Our system automatically detects events of interest, locates their positions in the VE relative to the user, and introduces diegetic stimuli that draw the user's gaze towards the events of interest. We realize an implementation of our event-driven approach through an immersive narrated tour application. **[brief description of how our methods]** Results of a user study evaluation show that our method is able to reliably draw users attention towards the correct objects of interest, and leads to improved information retention when quizzed about the scene content afterwards. To summarize, our main contributions are:

-

2 RELATED WORK

related work

2.1 Attention Guidance in Virtual Environments

In the realm of virtual reality (VR), attention guidance methods have been pivotal across various domains, including films and games, aiming to direct users' focus towards their intended targets [24]. One method, as demonstrated by Nielsen et al. [], involves forced rotation, altering users' perspectives by rotating their virtual bodies toward relevant content, while allowing users to freely move their heads. Subsequently, Lin et al. [] conducted a comparative study pitting this technique against an arrow-based approach, revealing user preference for auto-rotation in situations demanding extensive head movement. Nonetheless, it's worth noting that virtual rotation can have adverse effects on perceived immersion.

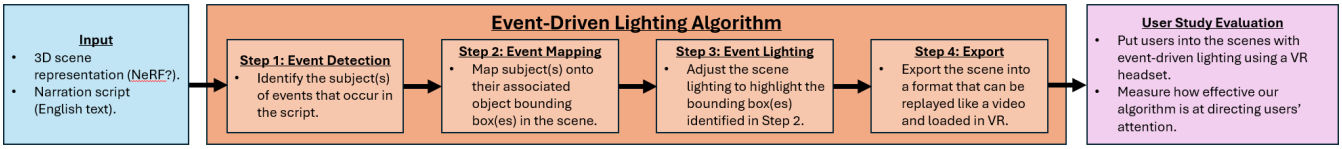


Figure 2: **E**vent-Driven Lighting Pipeline.

In pursuit of enhanced immersion, Gugenheimer et al. [] introduced SwiVRChair, a chair that physically reorients a user's real body toward relevant content, resulting in a significantly more immersive experience, devoid of simulator sickness. However, this method necessitates additional hardware for implementation.

Another strategy involves manipulating the virtual environment itself. Smith et al. [] employed blurred and non-blurred areas in videos to redirect viewers towards regions with minimal spatial blur when the rest of the image is blurred. Building upon this concept, Hata et al. [] used Gauss filters to determine thresholds at which blur is imperceptible without sacrificing guidance. Subsequently, Grogorick et al. [] implemented blurring within a virtual environment, testing it alongside a head-mounted display and a real-time video projection system within a full-dome setup. Among these configurations, blurring in conjunction with the head-mounted display was perceived as the most effective, albeit it's worth noting that blurring, while functional, may have adverse effects on immersion.

Various techniques have been explored to highlight pertinent content in VR. These include methods utilizing different shapes like arrows [] or halos [], as well as techniques positioned directly at the relevant content, such as Subtle Gaze Direction (SGD) [2, 12] and DeadEye []. SGD achieves its effect by modulating luminance and warm/cool flickering to alter the image area, while DeadEye draws attention to specific objects by rendering them in one eye only. However, it is essential to acknowledge that all of these techniques may potentially have a detrimental impact on immersion as they are not considered diegetic.

More recently, the HiveFive approach introduced a novel method that simulates swarm motion through particle effects to provide directional cues [16]. A comparative study demonstrated that this method yielded rapid response times while minimally affecting perceived immersion. Notably, the evaluation focused on scenarios where the target remained within the user's field of view, eliminating the need for head rotation.

- [9]
- [8]
- [21]

2.2 Automatic Event Detection

The event detection problem is a well known problem in natural language processing. The central challenge is to extract descriptions and timing information of occurances in the text from natural language alone. Events are context specific, for example in the context of a news headline, the event describes a real world interaction. In our system each event begins when a new entity in the scene becomes relevant to the user for some reason. This variety makes existing event detection systems like [17] fail in our use cases. Like [11], we investigate ChatGPTs capability to extract textual events given sufficient prompting. Event detection in video is another studied problem, [26] used event detection from video to aid users in aligning audio and visual qeueus. Event extarction from Audio is sparesly studied [28], in our experiments we find that the relative availability of text data, and the prevalence of easy to use text to speech [23] make textual event extraction sufficient, though improving timing information on event extraction from the audio itself should be explored.

- [26]

2.3 Automatic Content Creation

- 3D asset generation - Texture generation - Video generation - Automatic video editing - Other

2.4 Radiance Field Relighting

Radiance fields (neural and otherwise) [**?**, 10, 15, 20] are a widely popular technique for reconstructing and modeling real-world scenes from multi-view image captures. Such methods achieve remarkable view-synthesis results by representing the scene as volumetric color and density fields, yet they do not disentangle scene lighting from material and geometry, and therefore cannot be used directly for relighting. To address this, many works aim to decompose the radiance field into separate geometry, lighting and material components, by taking into account reflectance and material models in the radiance field optimization formulation [5, 13, 18, 25, 27].

3 Preliminaries

3.1 Cinematic Lighting Principles

Three-point lighting is a classic cinematography technique to illuminate a subject, and consists of the following key components [6]:

- **Key light** is the dominant or "main" light from in front of and/or above the subject (people or objects) of the scene.
- **Fill light** is (usually soft) light that fills in shadows which are not illuminated by the key light.
- **Backlight** illuminates the subject from behind and above (with respect to the camera viewing angle). Its purpose is often to create visual separation between the subject and the background.

The ratio between key to fill light defines the *contrast ratio*. Depending on this ratio, the lighting style as a whole is often characterized as being **high key**, i.e. bright and relatively shadowless, or **low key**, i.e. dark and shadowy.

A distinction can also be made between the *quality* of a particular light source [6]: **hard lighting** is defined as a small light source that casts sharp, well-defined shadows, while **soft lighting** comes from a large, diffused source that creates soft, fuzzy shadows. In a computer graphics lighting context, examples of the former include *point* and *spot* light sources, while examples of the latter includes *area* and *infinite area* sources (see definitions in Section 3.2.1).

The definitions above provide a basic working toolbox from which we will devise our strategy for lighting the scene to guide the user's attention in Section **??**. Though this only reflects a small subset of the vast array of established cinematographic concepts and techniques, we consider these to be fundamental building blocks from which more sophisticated lighting schemes may be derived, and defer the motivated reader to [1, 4, 6, 19] for more exhaustive background on cinematography principles.

3.2 Lighting in Computer Graphics

3.2.1 Physically-based Rendering

The rendering equation for reflective surfaces [14] defines outgoing radiance at surface point $\mathbf{x}$ in direction ω_o in terms of the emitted radiance L_e and incoming radiance L_i from all directions:

$$L_o(\mathbf{x},\omega_o) = L_e(\mathbf{x},\omega_o) + \int_{H^2} f(\mathbf{x},\omega_i,\omega_o) L_i(\mathbf{x},\omega_i) |cos\theta_i| d\omega_i$$

where θ_i is the angle between ω_i and surface normal $\mathbf{n}$ at point $\mathbf{x}$.

The **bidirectional reflectance distribution function** (BRDF), denoted in the rendering equation as $f(\cdot)$, describes the light scattering properties of the surface material. In particular, for a pair of directions ω_o and ω_i, the BRDF defines the ratio of reflected differential radiance to the differential irradiance [22]:

$$f(\mathbf{x},\omega_i,\omega_o) = \frac{dL_o(\mathbf{x},\omega_o)}{dE(\mathbf{x},\omega_i)}$$

Monte Carlo path tracing [14] is an unbiased, sampling-based algorithm for computing global illumination of a scene, and is based on defining the above rendering equation in a recursive manner to simulate multiple bounces of light. In this work, we will conduct our re-lighting with direct illumination only, however it is straightforward to extend to support global illumination.

3.2.2 Emitter Types

Here we briefly review the most common types of light sources, which will serve as our lighting primitives when it comes time to relight the scene for attention guidance. Note that physically-based rendering [22] follows the geometric optics model to describe light and light scattering, therefore the contribution from multiple light sources in the scene simply amounts to a linear combination.

- A **point light** at position $\mathbf{p}$ with radiant flux Φ (a.k.a. power) emits the same amount of illumination in all directions, and thus its *irradiance* E, or area density of flux arriving at a surface point $\mathbf{x}$, is defined as:

  $$E = \frac{d\Phi}{dA} = \frac{\Phi \cos(\theta)}{4\pi r^2}$$

  where $r = \|\mathbf{x}-\mathbf{p}\|$, the distance between the point source and surface point, determines the squared distance falloff of the energy arriving at the surface, and the $\cos(\theta)$ term accounts for *Lambert's law*, where θ is the angle between the surface normal and direction to the light source. To obtain radiance L_i as it is expressed in the rendering equation, we must also account for the fact that radiance is defined with respect to the surface area perpendicular to ω, rather than perpendicular to the surface, i.e. $dA^{\perp} = dA\cos(\theta)$. Finally, accounting for irradiance with respect to solid angle implies a visibility function $V(\cdot)$ between the surface point and the light source, which equals 1 if there is no occlusion, and 0 otherwise:

  $$L_i(\mathbf{x},\omega_i) = \frac{d^2\Phi}{dA^{\perp} d\omega_i} = \frac{\Phi}{4\pi r^2} \cdot V(\mathbf{x},\mathbf{p}) \tag{1}$$

- A **spot light** behaves similarly to a point light, except that it only casts light within a limited cone of directions rather than uniformly in all directions. This cone along with its *falloff* is defined by two angles: θ_{start}, the angle at which the falloff starts, and θ_{end}, the total width of the cone (each with respect to the vector along the center of the spotlight cone). For angles inside of the falloff start angle, the intensity of light is undiminished. Then the radiance L_i is defined as:

  $$L_i(\mathbf{x},\omega_i) = I(\theta) \cdot \frac{1}{r^2} \cdot V(\mathbf{x},\mathbf{p}) \tag{2}$$

  where

  $$I(\theta) = \begin{cases} c & \theta < \theta_{start} \\ t^2(3-2t)\cdot c & \theta_{start} \le \theta \le \theta_{end} \\ 0 & \text{otherwise} \end{cases}$$

  $$t = \frac{\cos(\theta) - \cos(\theta_{end})}{\cos(\theta_{start}) - \cos(\theta_{end})} \tag{3}$$

  θ is the angle between ω_i and the vector along the spotlight cone center, and c is constant defined on the RGB spectrum, assuming it satisfies that integrating the spot light intensity I over all directions ω gives the light's total power, Φ.

- Area Light

  $$L_i(\mathbf{x},\omega_i) = \frac{\Phi}{A}\frac{\cos\theta_A}{r^2} \cdot V(\mathbf{x},\mathbf{p})$$

  where A is the total area of the emitting shape, and θ_A is the angle between the area light normal $\mathbf{n}$ and $-\omega_i$, the vector from a point sampled on the area source to the surface point (i.e. $\cos\theta_A = |-\omega_i \cdot \mathbf{n}|$). **TODO: a bit more description.**

- Directional Light

- Infinite Area Light (HDR Skydome)

4 Event Detection

Definition of an event: An event is some change in the state of the scene, or external data.

From a script, the start of an event is the moment when a subject becomes part of what the speaker is saying. For instance, if the narrator begins describing a lizard in a nature documentary, that marks the beginning of the event. The subject of the event is "Lizard" Descriptions that follow the Lizard may also be part of the event.

4.1 Definitions & Terminology

In this section we provide a precise definition of *events* and introduce terminology that is used throughout this paper to describe how our system detects events and re-lights the scene to guide users' attention. Since our motivating application is immersive content that users can interact with, our notion of events is defined with this use-case in mind.

Let S be a scene (i.e. virtual environment) that is populated with objects O and agents A, such that $S = \{O \cup A\}$. Depending on the scene representation, O may be a set of polygonal meshes, point clouds, or volumetric fields (?). In this work, since we represent scenes as NeRFs, O is a set of bounding boxes o_i that delineate the locations of different objects in the scene. We define $U = \{u_1, u_2, ..., u_j\}$ as the set of users that are immersed in S. In this work we only consider single-user VR, so $|U| = 1$. We define a timeline T as an ordered set of timesteps $T = \{ t_0, t_1, ..., t_n \}$. The state S_k describes the configurations (positions and orientations) of all objects and users in the scene at time $k \in [0,n]$.

Informally, an event is anything that happens in the scene that the user should care about and that we want the user to attend to. Since the user must be attending to something in the scene, an event must be associated with at least one object in the scene at some point in time. Formally, we can define an event $e = (O^*, k, d)$ as a 3-tuple of objects $O^* \in S$ that we want the user to attend to at time $t = k$ for a duration of d timesteps. Since this notion of events is very general, the exact occurrences in the scene that qualify as events will depend

on the context of the VR application and what the author wants the user to pay attention to. In the next section, we provide concrete examples of how to map our definition of events onto occurrences that are likely to be encountered by a user in a VR application.

[todo: don't know if this framework will work for audio events. some audio events have an ambiguous location (e.g. thunder)]

[todo: do we need to introduce the notion of a light in this section, since the light is how we actually guide users' attention to the events?]

4.2 Types of Events

The definition of the event lies in a wide range, based on different scenarios where we can detect an event (detect, track, predict), we roughly divide the event into four types:

- Actor actions from video (e.g. gaze, interaction with the scene)
- Object motion/collision of scene elements
- Text descriptions (movie script, narration)
- Audio

The detection of each of these four types varies from domain to domain. In text, we can detect the subject of a sentence changing as the most straight forward example. In audio, the start of a new source playing, or a particular kind of sound starting is an event. In video, an agent entering the screen, or a particular sort of motion can be detected.

In this paper our goal is to automate lighting using events derived from a script. This practical example is especially of interest because we are generating 3D dynamic content using text as the input format. For this reason we focus on the defintions and detection of events from text.

4.3 Extracting Events from Script

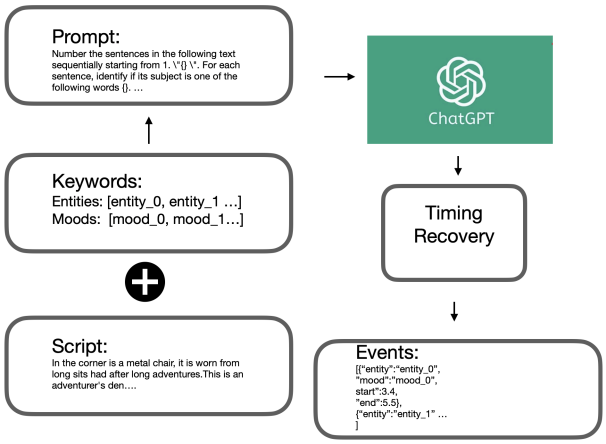
Using events to relight a scene is a two phase process. First, events must be generated in some way, we use scripts since text is an extremely practical modality for event extraction. Once events are generated they can be used by any compatible system. We demonstrate this with a game-like scene in Unity, and a Nerf based scene.

Solving Event textual extraction is a difficult problem [17]. Our approach automatically collects entity and mood information from the script using a keyword based prompting strategy for ChatGPT. We provide lists of keywords for each type of information we wish to extract, in this case entity, and mood. These fields are grouped with their associated sentences and processed further with a rudimentary timing estimation system. This simple pipeline produces events with suitable timing and mood information for content generation in event based systems. Another approach using purpose built IE models [17] was also tried, however the domain gap between the news headline based event extraction work, which has entites as nations and people does not perform well on scripts where household objects are the entites of interest.

5 Event-based Scene Relighting

5.1 Lighting Design for Guiding User Attention

Using the lighting principles outlined in Section 3.1, we describe a lighting design method for illuminating the current subject in the scene (see Figure 4). First, the positioning of lights is dependent upon the camera viewing angle toward the subject, as this determines whether a light source is perceived as a back light or keylight to the viewer. Next, we strive for simplicity in our lighting design, such that it effectively accentuates the subject over a wide range of scenarios. Therefore, we use an area light as the key, placed directly above the subject bounding box, in order to achieve s naturalistic



soft light source. Then, a back light is then cast from behind the subject with respect to the viewing angle, at an elevation of θ_{back} above the horizontal plane. Finally, the effect of these light sources, on their own, and together, are integrated to guide the user attention. An example illustrating these steps are shown in Figure 5.

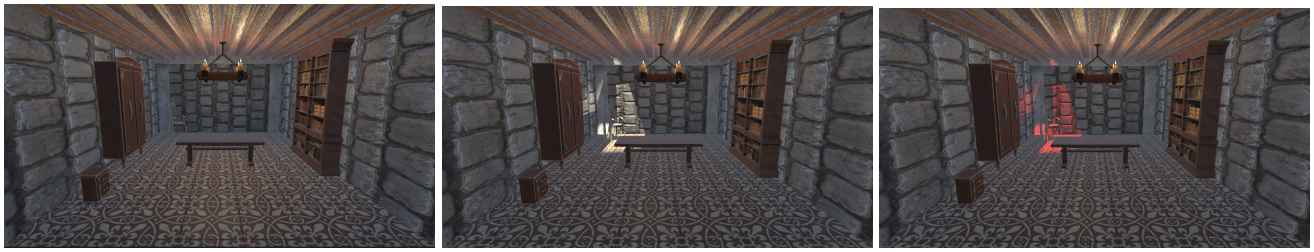
In spite of our formal user evaluation of the effectiveness of our design choices in Section 6, we acknowledge that lighting design can be a subjective art form, thus we note that alternative approaches for lighting design may be similarly effective at guiding user attention or fulfilling other objectives, such as appearing natural or motivated by the story context, and an exhaustive comparison of such alternatives is a promising direction for future work.

5.2 3D Event/Object Detection

TODO: Talk about LERF If we get it in In game-like environments, the concept of an event, and its mapping to 3D is well understood. Events are used extensively in game engines, due to their intuitive mapping to things like keystrokes or collisions, as well as the benefits they offer architectually, namely multi-threading, system de-coupling, and replayability of game scanarios. For this reason our method integrates seemlessly into state of the art game-engines in industry. We demonstrate this in our game like environment where our LLM derived structured events are read into a simple system which walks the scene hierarchy to select entities by name. Then a relighting system can execute arbitraty code on the data of the event, and any other information stored on the entity in the scene to change the lighting. The effect is that with very little user effort the mood, and clarity of the scene is changed. In our example we have a narrator describing a room. The room has furniture in it, and the narrator describes a few peices. Our system extracts names of entites from the script, along with the mood of the sentences in which they are described as an event stream. This stream is consumed by a very simple less than 200 line unity scene controller, which then automatically relights the scene with entity and mood information.

5.3 Radiance Field Relighting

A primary challenge in relighting a volumetric radiance field representation of a 3D scene lies in decomposing the volume such that geometry, material, and lighting are disentangled. In this work, we leverage TensoIR [13] as our baseline radiance field decomposition method, however we wish to emphasize that our relighting system is conceptually agnostic to the specific choice of decomposition method. Assuming that the volume can be decomposed into geometry (which determines visibility of light sources from surface points), material (which defines reflection and scattering behavior of light), and light sources, then relighting the radiance field can be achieved with standard techniques from ray tracing, as rendering is not restricted to the differentiable volume rendering scheme of canonical



(a) Default scene lighting (b) Spot light driven by events (c) Spot light with mood color

Figure 3: **Effect of automatic lighthing on scene clarity and feel** In the first example, the narration has no impact on scene lighthing. In the second, while the narration is focused on the chair it is illuminated with a spot light. In the third, the chair is illuminated and the color of the light is dictated by the sentiment with which the narrator refers to the chair.

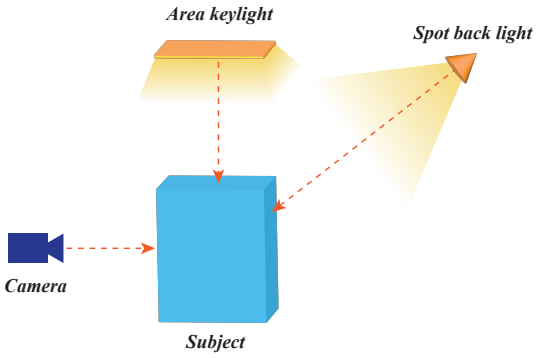
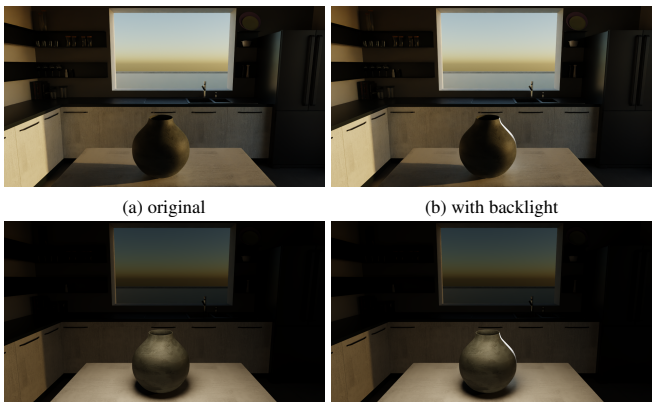


Figure 4: **R**ecipe for illuminating the subject.



(a) original (b) with backlight

(c) with top light (d) with top and backlight

Figure 5: **Effect of illuminating the vase (subject).** These renders, created using Blender [7], compare the effect of adding a *key light*, and *backlight*. The backlight creates a bright rim around the subject, highlighting it in addition to separating it from the background.

radiance field reconstruction methods like NeRF [20], which has built-in assumptions, modeling the scene as color and density fields.

During rendering, TensoIR [13] computes the radiance as a sum of indirect and direct illumination contributions, where the direct illumination term represents the direct contribution of the HDR skydome upon a surface, and subsequent bounces that contribute to indirect illuniation are modeled by volume rendering rays starting from surface points into the scene:

$$L_i(\hat{\mathbf{x}}, \omega_i) = V(\hat{\mathbf{x}}, \omega_i) L_d(\omega_i) + L_{ind}(\hat{\mathbf{x}}, \omega_i) \tag{4}$$

In order to incorporate point, spot and area lights into this formulation, we can simply modify the direct illumination $L_d(\omega_i)$ at the surface point by randomly sampling between the HDR Skydome light source which TensoIR assumes, and the other types of emitters that we have defined. A simple way to achieve this is to uniformly randomly select a light source $S \in \{S_i\}_{i=1}^{n}$ for every sampled path, and compute the corresponding contribution of direct illumination $V(\cdot)L_d(\cdot)$ via Eqs. (1, 2, 3). In expectation over multiple ray samples per rendered pixel, this will result in an unbiased estimate of the direct illumination.

Note that this formulation is practical from a computation perspective, yet it does not factor relighting into the indirect lighting computation. This would be achieved easily by sampling multiple bounces of the ray to explicitly compute global illumination. In this case, we could simply remove the indirect contribution term L_{ind} in Eq. 4, since the indirect lighting is computed explicitly "online".

See Figure 6 for results of adding point, spot, and area lights to the rendering of a decomposed radiance field. These are the lighting primitives that will serve as the basis to construct our scene re-lighting, as they are already well known and ubiquitously used representations of light sources in computer graphics, can achieve a wide range of cinematic lighting effects, and are sufficient to emphasize important regions/entities within the scene.

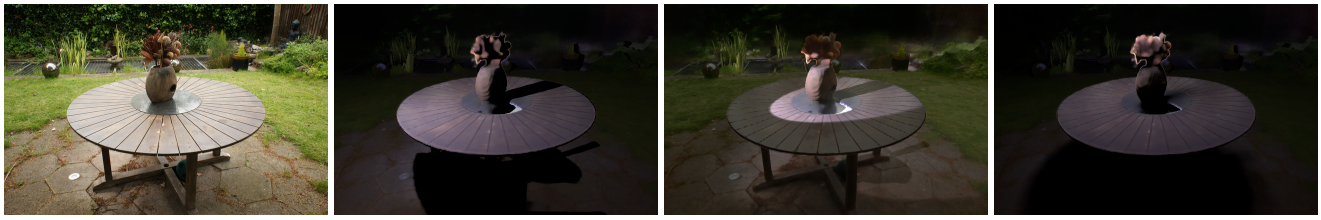
6 Evaluation

experiment design, setup, participants

3 experiment conditions:

- event-driven lighting: our contribution. hopefully it will be both immersive (does not break presence) and effective
- subtle gaze direction: An immersive attention-direction method. hopefully has similar levels of immersion as our method, but hopefully not as effective/not preferred by users?
- GUI arrows: non immserive method to overtly guide user attention. baseline/control condition

experiment design: user is placed into a virtual environment and there is a narration playing. The scenario is something like a history lecture or a museum tour of a historical room (e.g. anne frank's living



(a) Original camera image (b) Point light (c) Spot light (d) Area light

Figure 6: **Direct illumination relighting results from various emitter types.** The Garden scene [3] is represented as a TensoIR [13] decomposition of the scene radiance field. Notice the effect of the emitter type upon the appearance of shadows in the scene. (a) and (c) have hard shadows, while the area light of (d) permits soft shadows.

room). The narration discusses various objects in the room one-by-one. Each introduction of a new object in the room is a new event, which triggers our system to udpdate the lighting to highlight the new object. The user may be given some auxiliary task to force them to try to pay attention to the objects (e.g., they will have a small quiz at the end of hte experiment where they answer questions about the objects/the narration content). user will experience one lecture for each condition (cannot repeat a lecture, since the user will learn the locations of objects from trial to trial). pairing between conditions and rooms will be counterbalanced across participants. To measure the effectiveness of the techniques, we will measure quantitative data such as response time, gaze trajectory, and qualitative data through questionnaires regarding immersion and perceived presence.

make sure in the experiment that we tell users that they will need to complete an information recall/scene understanding task after each trial

can we make one of the tasks take place in a kitchen? since it has objects with cool object properties (reflective surfaces like knives etc)

another experiment (ming's suggestion): which lighting do they prefer?

what's the best application of this tool? ming votes for some kind of game application. maybe we could do some kind of simple VR target shooter game(??)

7 Results and Discussion

results and discussion

8 Limitations and Future work

todo

9 Conclusion

todo

References

[1] J. Alton. *Painting with Light*. Macmillan, 1949.

[2] R. Bailey, A. McNamara, N. Sudarsanam, and C. Grimm. Subtle gaze direction. *ACM Transactions on Graphics (TOG)*, 28(4):1–14, 2009.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

[4] J. Birn. *Digital Lighting & Rendering*. New Riders Pub., 3rd ed., 2013.

[5] M. Boss, V. Jampani, R. Braun, C. Liu, J. T. Barron, and H. P. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[6] B. Brown. *Cinematography : Theory and Practice*. Routledge, New York, NY, 2021.

[7] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[8] F. Danieau, A. Guillo, and R. Doré. Attention guidance for immersive video content in head-mounted displays. In *2017 IEEE Virtual Reality (VR)*, pp. 205–206. IEEE, 2017.

[9] M. S. El-Nasr, A. Vasilakos, C. Rao, and J. Zupko. Dynamic intelligent lighting for directing visual attention in interactive 3-d scenes. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(2):145–153, 2009.

[10] Fridovich-Keil and Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[11] J. Gao, H. Zhao, C. Yu, and R. Xu. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*, 2023.

[12] S. Grogorick, M. Stengel, E. Eisemann, and M. Magnor. Subtle gaze guidance for immersive environments. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 1–7, 2017.

[13] H. Jin, I. Liu, P. Xu, X. Zhang, S. Han, S. Bi, X. Zhou, Z. Xu, and H. Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86. ACM, 1986.

[15] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[16] D. Lange, T. C. Stratmann, U. Gruenefeld, and S. Boll. Hivefive: Immersion preserving attention guidance in virtual reality. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–13, 2020.

[17] Y. Lin, H. Ji, F. Huang, and L. Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7999–8009, 2020.

[18] A. Mai, D. Verbin, F. Kuester, and S. Fridovich-Keil. Neural microfacet fields for inverse rendering, 2023.

[19] J. V. Mascelli. *The Five C's of Cinematography: Motion Picture Filming Techniques*. Silman-James Press, 1998.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[21] N. Norouzi, G. Bruder, A. Erickson, K. Kim, J. Bailenson, P. Wisniewski, C. Hughes, and G. Welch. Virtual animals as diegetic attention guidance mechanisms in 360-degree experiences. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4321–4331, 2021.

[22] M. Pharr, W. Jakob, and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. The MIT Press, 4th ed., 2023.

[23] J. F. Pitrelli, R. Bakis, E. M. Eide, R. Fernandez, W. Hamza, and M. A. Picheny. The ibm expressive text-to-speech synthesis system for american english. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1099–1108, 2006.

[24] S. Rothe, D. Buschek, and H. Hußmann. Guidance in cinematic virtual reality-taxonomy, research status and challenges. *Multimodal Technologies and Interaction*, 3(1):19, 2019.

[25] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7495–7504, 2021.

[26] J. Sun, L. Deng, T. Afouras, A. Owens, and A. Davis. Eventfulness for

interactive video alignment. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

[27] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 2021.

[28] X. Zhang, Z. Wang, and P. Li. Multimodal chinese event extraction on text and audio. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2023.