Graphics are disabled. Document is incomplete.
Enable graphics by using `\graphicstrue` in `main.tex`.

Master's Thesis

F3 Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

SIMR

Simulating the phenomena of altered states of consciousness using virtual reality

Jakub Hlusička

2021–2022

Supervisor: Ing. Josef Kortan

Diplomová práce

F3 Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

SIMR

Simulace fenoménů pozměněných stavů vědomí pomocí virtuální reality

Jakub Hlusička

2021–2022

Školitel: Ing. Josef Kortan

Abstract

TODO

Acknowledgements

TODO thank sci-hub

—Jakub Hlusička

"Not everything that is faced can be changed, but nothing can be changed until it is faced."

—James Baldwin, As Much Truth as One Can Bear (1962)

Contents

1 | Implementation

The objective of this part of the project was to implement an immersive replication of altered states of consciousness (ASCs) induced by classical psychedelics. In order to achieve a high degree of immersion, our solution was designed to be intended for immersive virtual reality (VR) systems with head–mounted displays (HMDs). Our solution will be referred to with *"the application"* or *"our application"* for the rest of this document.

To represent the ASCs of classical psychedelics objectively, we had to resort to modelling only the "perceptual level" stage of the psychedelic experience (as seen in figure ??), as further stages require subjective personalization of content, and aspects less suitable for replication via immersive VR, such as cognitive effects and the suppression of the *phenomenological ego*.

1.1 Design of the Application

The application was designed primarily for the evaluation of the implemented replication. The development of the application consisted of 3 distinct parts:

1. **The environment**: A virtual scene that should look as realistic as possible.
2. **The replication**: Implementation of the effects themselves.
3. **Adaptation for testing**: Getting the application ready for a study, that might measure the impact of the replication on the human mind.

1.1.1 Safety

In order to ensure our application's users safety, we have consulted the *Recommendations for good scientific practice and the consumers of VR-technology* (Madary and Metzinger 2016). The application was developed according to these recommendations.

Mainly, we don't expect the developed application to have lasting traumatic effects on the users; instead, we believe that this medium may be a suitable way to explore aspects of psychedelic-induced ASCs while minimizing those risks.

Further, we've taken safety and intuitiveness into account while designing the controls and choosing a suitable testing area for experimentation with VR (the

"VR play space").

Finally, the application must be automated, so that the administrator may assist the user and ensure their safety during the usage of the application.

1.1.2 Interaction

Interaction with the scene via hand-held controllers was removed entirely, as we felt that the currently available consumer VR technology does not implement a realistic, consistent, universal and intuitive solution for interaction with the virtual scene. For example, in VR applications, interactions are usually implemented so that if a user takes a hand-held controller to a dynamic physics-enabled object, they may be able to pick it up by pressing or holding a trigger on the hand-held object, which makes the object stick or snap to the virtual representation of the controller in the scene. While this solution may be suitable for VR games, it is still understood as a simplification.

As an alternative, one may consider using force feedback haptic gloves, and given a sufficient physically based simulation, it may be possible to implement realistic interactions with virtual objects. However, even such gloves apply force feedback only to the fingers and not the entire body, making it impossible to, for example, lean against virtual objects.

In any case, no such force feedback haptic gloves were available to us for this project, and so interaction was entirely foregone, in the interest of keeping the simulation focused mainly on the replication, rather than an unrealistic implementation of interactions.

1.1.3 Virtual Scene Creation

Given the goal of creating as realistic of a scene as possible, as well as no financial budget for this project, we ended up choosing Unreal Engine 4 (UE4)[1] as the game development engine to develop our VR application with. UE4 is free to use for projects with a lifetime gross revenue below $1 million USD, and we have no plans to monetize it. Additionally, the choice of UE4 makes it possible to use the Quixel Megascans[2] 3D asset library for free within UE4, due to special licensing as a result of the acquisition of Quixel by Epic Games[3], the developer of UE4.

The virtual scene was created with the intended *VR play space* in mind, which was measured to be about 3.5×3.5 m^2 large. The virtual scene contains visual cues of the *play area* borders in the form of 3D assets; specifically, the *play area* is surrounded by a railing and tall rock, communicating to the user, that these objects should not be passed through.

The choice was made to create an outdoor scene, as the surrounding nature might

[1] https://web.archive.org/web/20220514231756/https://www.unrealengine.com/en-US
[2] https://web.archive.org/web/20220514233901/https://quixel.com/megascans
[3] https://web.archive.org/web/20220514235540/https://www.epicgames.com/site/en-US/home

provide a more pleasant environment than an indoor scene. However, our implementation is in no way limited to outdoor scenes only.

At first, we attempted to create a forest scene, but quickly ran into performance issues while trying to render a densely populated forest on a HMD, which requires at least 2 views rendered at typically higher resolutions than a regular desktop screen, ideally with at least 90 FPS (the native refresh rate of the HMD). Delivering a consistent framerate is a requirement, as low framerates and stuttering may cause motion sickness.

It was then decided to abandon the idea of a forest scene and, instead, use a high dynamic range imaging (HDRI) panoramic photograph as a background (hereinafter "panoramic background") for the scene. It is important to note, that a panoramic background has no depth information. This drawback can be mitigated by making only the very distant parts of the panoramic background visible to the user, so that the illusion of the panoramic background being realistic is not broken. The illusion relies on the fact that binocular disparity is low for distant objects.

Close parts of the panoramic background can be hidden with 3D assets suitable for the environment. To minimize the area that needed to be hidden, we have chosen a mountainside panoramic background (see figure 1.1).

Figure 1.1: The chosen panoramic background "Cannon"[4] available on Poly Haven[5].

The final scene contains a flat patch of grass and other low foliage the size of the *play area*, containing a wooden bench with some gardening tools. The grass patch is surrounded with rock formations and a rocky stairway leading towards it. Beyond the railing, there is a nice view of the sea cove.

Figure 1.2: A preview of the resulting scene.

1.2 Implementation of Replications

The following replications are modelled after surveys of the phenomenology of psychedelic states (Preller and Vollenweider 2016; Kometer and Vollenweider 2016) and personal reports (Kleinman, Gillin, and Wyatt 1977).

1.2.1 Spatial Effects

The spacial effects of this section are a form of a vertex shader, or part thereof. UE4 has a concept of so-called "materials" – assets that can be applied to meshes

[4] Released by Greg Zaal under the CC0 1.0 (public domain) license.
[5] https://web.archive.org/web/20220515010919/https://polyhaven.com/a/cannon

to control the visual look of 3D assets. UE4 materials are defined using a built-in visual programming node graph. While this approach of specifying graphics processing unit (GPU) shader logic allows for tighter integration with UE4's rendering engine and its lighting model, it makes it difficult to use conventional shading languages such as OpenGL Shading Language (GLSL) or High Level Shading Language (HLSL). The usage of conventional shading languages is sometimes desirable, because the provided material node graph editor cannot, by design, express some control flow constructs, such as loops.

However unwieldy, it is possible to use HLSL code in material graphs, in UE4. The directory of our custom HLSL shader files must be properly registered in the engine via an engine plugin (Alessa Baker 2021). The shader files can then be referenced from within "custom" nodes of the material node graph editor.

1.2.1.1 Depth Perception Distortion

This replication simulates the distorted perception of depth, micropsia, and macropsia (Fischer et al. 1970; Dittrich 1998; Hill, Fischer, and Warshay 1969; Hill and Fischer 1973).

Our replication of the depth perception distortion is implemented as a world-space vertex shader. Even though certain distortions in screen-space may also result in the distortion of depth perception, we chose our approach to achieve better control over the shape of the rendered geometry.

1.2.1.1.1 First Attempt Our first attempt made use of a self-similar, bijective and continuous function f to modify the distance of vertices from the HMD. The self-similarity is required to ensure that no self-intersection of geometry occurs, after the offset has been applied.

$$f(r) = a^{\frac{1}{\pi}\sin(\pi \log_a r)+\log_a r} \tag{1.1}$$

Where $a \in (1; +\infty)$.

The interesting property of self-similarity results in the maximum offset being directly proportional to the distance of the HMD. The graph of the function can be seen in figure 1.3.

Figure 1.3: First attempt at distorting depth perception, unused in the final application. The function f from equation 1.1 modifies the distance of vertices from the HMD. Parameter $a = 1.5$.

Although this first iteration resulted in visually enticing results, we also noticed that it caused motion sickness in VR, particularly during the user's movement. We believe the motion sickness was caused by clusters of geometry moving with

the position of the HMD, as if attracted to it, and caused a discrepancy between the user's vestibular system and the visual information.

Another disadvantage of this solution is its predictability and the synthetic look caused by its uniformity.

1.2.1.1.2 Final Solution In order to break up clusters of similarly affected geometry, we decided to use procedural noise. Procedural noise has been used for the generation of textures (Perlin 1985), and would be suitable to create a less predictable, more chaotic effect.

There are various kinds of procedural noise used in computer graphics. We found Simplex noise (Olano et al. 2002) to be suitable for our use-case, particularly for its $\mathcal{O}(n^2)$ time complexity in n dimensions, lack of directional artifacts (visual isotropy), and a well-defined continuous gradient.

Let's denote the simplex noise sampling function as $s\colon \mathbb{R}^n \to \mathbb{R}$.

Now that we have a way to sample the noise, we could try to use the noise sample (possibly scaled by a constant) as the offset of the distance of each vertex to the HMD, as shown in the previous section. To retrieve the sample, we can use:

$$s' \coloneqq s\left(\begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix}\right) \tag{1.2}$$

where s' is the resulting sample, x, y and z are spacial coordinates of any point in space (typically the position of a vertex), t is the current time in seconds, and f_s and f_t are space-wise and time-wise frequencies of the noise, respectively.

With the addition of the time coordinate, resulting in 4-dimensional noise, the sampled offset value will change over time. This helps break up the uniformity and predictability of the previous solution.

Unfortunately, we have lost one important property of the previous solution: The fact that the maximum offset was directly proportional to the distance from the HMD. We could try to reintroduce proportionality naively by multiplying the sample by the distance r:

$$s' \coloneqq r \cdot s\left(\begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix}\right) \tag{1.3}$$

By this modification, we have effectively changed the amplitude of the noise, yet the frequency remains the same – and uniform. If we were to use this sample s' as the distance offset, we would cause distant geometry to self-intersect. Therefore, a different solution is needed.

For this task, we may use fractional Brownian motion (fBm). Before we go into how fBm can help resolve this issue, let's briefly describe how it is used in the synthesis of self-similar noise. fBm is a technique of combining layers of noise, while varying their amplitudes and frequencies.

$$s' := \frac{1}{k}\sum_{i=0}^{k-1} g^i \cdot s\left(l^i \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + i\vec{o}\right) \tag{1.4}$$

In equation 1.4, we combine $k \in \mathbb{N}$ layers of simplex noise. The parameter $g \in \mathbb{R}$ (gain) changes the amplitude of each layer, and the parameter $l \in \mathbb{R}$ (lacunarity) changes the frequency of each layer. Typically, $l = \frac{1}{g}$; in that case, we are generating so-called "pink noise". The parameter $\vec{o} \in \mathbb{R}^n$ is a coordinate offset applied to each layer, to break up symmetry (a different seed for each layer could also be used).

Now, back to our original problem of keeping the amplitude proportional to the distance from the HMD. Instead of using k layers of noise with indices $i = 0, 1, \ldots, k-1$, we can offset the indices by an integer value depending on the distance from the HMD. If we do so carefully, we will recover the direct proportionality. We define a real-valued offset $m\colon \mathbb{R} \to \mathbb{R}$:

$$m(r) = \log_a r \tag{1.5}$$

Where $a \in \mathbb{R}$ is a parameter; more on this later. Then, we can define the rounded-down integer part $j\colon \mathbb{R} \to \mathbb{Z}$ to offset the layer indices with:

$$j(r) = \lfloor m(r) \rfloor \tag{1.6}$$

$$s' := \frac{1}{k}\sum_{i=0}^{k-1} g^{i+j(r)} \cdot s\left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o}\right) \tag{1.7}$$

Equation 1.7 differs from equation 1.4 in that we have added an index offset $j(r)$, which is computed from the distance to the HMD r. Now we are able to offset the indices. However, this will result in noticeable discontinuities of the noise at the discontinuities of $j(r)$.

In order to remove these discontinuities, we shall introduce blending. Let us refer to the layers at indices $j(r), j(r)+1, \ldots, j(r)+k-1$ as "active layers", the layer at index $j(r)$ as "the first active layer" and the layer at index $j(r)+k-1$ as "the last active layer".

Let us define a weight function $w\colon \mathbb{Z} \times \mathbb{R} \to [0;1]$, which will attenuate the first and last active layer, while leaving other layers unaffected.

$$w(i,r) = \begin{cases} m(r) - j(r) = \{m(r)\} & \text{if } i = 0 \\ 1 - (m(r) - j(r)) = 1 - \{m(r)\} & \text{if } i = k \\ 1 & \text{otherwise} \end{cases} \tag{1.8}$$

Where $\{x\} = x - \lfloor x \rfloor$ is the upper fractional part of x.

$$\boxed{s' := \frac{1}{k} \sum_{i=0}^{k} w(i,r) \cdot g^{i+j(r)} \cdot s \left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o} \right)} \tag{1.9}$$

Besides multiplying each layer by the weight $w(i,r)$, we have also changed the upper index of the sum from $k-1$ to k, to account for the attenuation.

We have arrived at the derived general equation for blended fBm noise based on distance r with parameters $a, g, l \in \mathbb{R}; \vec{o} \in \mathbb{R}^n; k \in \mathbb{N}$. However, in general, this form does not satisfy our requirement of the amplitude being proportional to the distance r. In order for that to be true, we must set $a = g = \frac{1}{l}$.

The equation then expands to:

$$s' := \frac{1}{k} \sum_{i=0}^{k} w(i,r) \cdot a^{i+\lfloor \log_a r \rfloor} \cdot s \left(a^{-(i+\lfloor \log_a r \rfloor)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + \lfloor \log_a r \rfloor)\vec{o} \right) \tag{1.10}$$

$$w(i,r) = \begin{cases} \{\log_a r\} & \text{if } i = 0 \\ 1 - \{\log_a r\} & \text{if } i = k \\ 1 & \text{otherwise} \end{cases} \tag{1.11}$$

Finally, we may or may not want the frequency of the fourth coordinate to be influenced by lacunarity. We reached better results when lacunarity did not affect the time coordinate.

$$\boxed{s' := \frac{1}{k} \sum_{i=0}^{k} w(i,r) \cdot a^{i+\lfloor \log_a r \rfloor} \cdot s \left(\begin{bmatrix} a^{-(i+\lfloor \log_a r \rfloor)} f_s x \\ a^{-(i+\lfloor \log_a r \rfloor)} f_s y \\ a^{-(i+\lfloor \log_a r \rfloor)} f_s z \\ f_t t \end{bmatrix} + (i + \lfloor \log_a r \rfloor)\vec{o} \right)} \tag{1.12}$$

This is the final equation used to sample blended fBm noise based on the distance r with parameters $a \in \mathbb{R}, \vec{o} \in \mathbb{R}^n, k \in \mathbb{N}$. This equation is used in the application to displace vertices of scene geometry. The result is scaled to ensure that no self-intersections of scene geometry occur.

Figure 1.6: Computation of active layers based on the distance from the HMD. Active layers as solid black lines. Green areas correspond to the weight of the first and last currently active layer.

Figure 1.7: Amplitudes of active layers based on the distance from the HMD, for the case where $a = g = 1.5$. Active layers as solid black lines. The green line shows the direct proportionality of the maximum active layer amplitude and the distance from the HMD.

1.2.1.2 Visual Drifting

In this section, we describe the implementation of our replication of visual drifting, sometimes described as the "breathing" or "morphing" of objects (Díaz 2010; Kleinman, Gillin, and Wyatt 1977).

As with depth perception distortion, there are two ways of implementing this replication, depending on the coordinate system and stage of rendering affected – world-space and screen-space.

Ideally, the realism of the final implementation should not be broken by:

1. the rotation of the HMD;
2. the movement of the HMD up to regular walking speed;
3. the fast movement of objects in the scene.

We chose to avoid a screen-space implementation, because satisfying any of those requirements seemed very difficult. Nevertheless, we are confident that our world-space solution satisfies at least the first two of the requirements.

We can use procedural noise to offset vertices into arbitrary directions in 3 dimensional space, but we need to be careful so that we do not cause geometry to self-intersect. Thankfully, we have already solved this problem in the previous section, and we can use that noise function.

Let us consider the general equation 1.9 with $a = g = \frac{1}{l}$, and denote the right-hand side expression as a vector field $S\colon \mathbb{R}^n \to \mathbb{R}$, where $n = 4$:

$$S\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}\right) = \frac{1}{k}\sum_{i=0}^{k} w(i,r) \cdot g^{i+j(r)} \cdot s\left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o}\right) \tag{1.13}$$

Then, we have two ways of using this noise function to generate an offset vector $\vec{v} \in \mathbb{R}^3$.

Option one.
We compose the offset vector $\vec{v}$ from three samples of S with coordinate offset vectors $\vec{o}_0, \vec{o}_1, \vec{o}_2 \in \mathbb{R}^4$ to ensure different samples (different seeds would also work).

$$\vec{v} := \begin{bmatrix} S\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} + \vec{o}_0\right) & S\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} + \vec{o}_1\right) & S\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} + \vec{o}_2\right) \end{bmatrix}^T \tag{1.14}$$

Option two.
We use the first 3 components of the gradient of S as the offset vector $\vec{v}$.

$$\vec{v} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \nabla S\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}\right) \tag{1.15}$$

The second option seems much more elegant, does not require 3 samples, and may even yield nicer results. Option two was our choice.

Unfortunately, the distance from the HMD r is dependent on the x, y and z coordinates, which makes the gradient overly complex. For that reason, we have decided to simplify the gradient by treating the r parameter as a constant. We shall denote this simplification of the gradient ∇S as $S_G \colon \mathbb{R}^n \to \mathbb{R}^n$.

$$S_G\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}\right) = \frac{1}{k} \sum_{i=0}^{k} w(i,r) g^{i+j(r)} \nabla s \left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o} \right) \odot \begin{bmatrix} f_s \\ f_s \\ f_s \\ f_t \end{bmatrix} l^{i+j(r)} \tag{1.16}$$

$$= \begin{bmatrix} f_s \\ f_s \\ f_s \\ f_t \end{bmatrix} \odot \left(\frac{1}{k} \sum_{i=0}^{k} w(i,r) g^{i+j(r)} \nabla s \left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o} \right) l^{i+j(r)} \right) \tag{1.17}$$

The $\odot$ operator is the component-wise vector multiplication. For $g = \frac{1}{l}$, we could

simplify further:

$$S_G\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}\right) = \begin{bmatrix} f_s \\ f_s \\ f_s \\ f_t \end{bmatrix} \odot \left(\frac{1}{k} \sum_{i=0}^{k} w(i,r) \nabla s \left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o} \right) \right) \tag{1.18}$$

As we can see in figure 1.8, the direct proportionality of the amplitude and the distance r has been lost. We could have also noticed the $g^{i+j(r)}$ term being cancelled during the simplification step to equation 1.18. In order to recover it, we need to divide the right-hand side of equation 1.17 by $l^{i+j(r)}$.

$$S'_G\left(\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}\right) = \begin{bmatrix} f_s \\ f_s \\ f_s \\ f_t \end{bmatrix} \odot \left(\frac{1}{k} \sum_{i=0}^{k} w(i,r) g^{i+j(r)} \nabla s \left(l^{i+j(r)} \begin{bmatrix} f_s x \\ f_s y \\ f_s z \\ f_t t \end{bmatrix} + (i + j(r))\vec{o} \right) \right) \tag{1.19}$$

We denote this modulated version of the simplified gradient as S'_G. As seen in figure 1.9, the proportionality is recovered.

This is the final function we have used in our application to displace vertices of the scene geometry. To recapitulate the requirements we have placed upon ourselves to implement this replication, we required the realism not to be broken by the following actions.

- ☒ The rotation of the HMD. Does not influence the sampling in any way, only the location of the HMD is relevant. Success.
- ☒ The movement of the HMD up to regular walking speed. The movement of the HMD makes objects shift their active layers in a continuous, smooth way. Success.
- ☐ The fast movement of objects in the scene. Objects moving fast across the screen(s) might appear to morph erratically, significantly faster than the animation speed of still objects. Not ideal.

Regarding the last point, one might think to modulate the vertex offset by the object's inverse velocity to account for the erratic animation, however, that might result in the object intersecting with the rest of the scene – for example, with a static wall unaffected by the modulation.

We did not expect any fast-moving objects in the scene, and so this implementation was sufficient for our application.

1.2.2 Non-Spatial Effects

The following replications are implemented entirely as screen-space post-processing effects, that is, applied to the 2-dimensional texture which results from rendering the 3-dimensional scene. VR HMDs typically make use of multiple screens for stereoscopic rendering. The post-processing effects are applied to all of the textures corresponding to each screen in the HMD.

While UE4 does provide a way to create post-processing effects via "post process materials", these assets are limited by the design of the rendering engine and the lighting model. For example, it is not possible to define custom framebuffers for use in temporal post-processing effects.

Such features require the modification of the rendering dependency graph (RDG), UE4 graph-based abstraction designed to perform whole-frame optimization of the rendering pipeline. At the same time, the documentation of this feature is very limited, with the most reliable resources being the engine's code itself, and various blog posts on the internet.

Thankfully, we could make use of a boiler-plate example plugin (Ossi Luoto 2021) that showcases interaction with the RDG, which we modified for our purposes.

1.2.2.1 Visual Acuity Enhancement

Classical hallucinogens are known to cause alterations in the perception of elementary visual features, such as brightness, color saturation, and visual contrast (Heinrich Klüver 1942; Klüver 1966; Dittrich 1998; Díaz 2010; Siegel and Jarvik 1975; Fischer, Hill, and Warshay 1969), as well as in the perception of detail in the textures of objects (PsychonautWiki 2020). This may contribute to the sense of novelty typical during classical hallucinogen-induced ASCs.

1.2.2.1.1 Enhancement of Saturation and Contrast

The implementation of the enhancement of saturation and contrast is done entirely with the use of in-engine features of UE4. UE4 makes it possible to define a global post-processing volume (a volume spanning the entirety of the scene), that applies select basic post-processing effects to the rendered views. These effects include modifiers for the saturation and contrast of the rendered views.

Besides saturation and contrast, a modifier for brightness is also available, although we did not end up using it, because the dynamic range of our VR headset (the *HTC Vive* and the *HTC Vive Pro*) was not sufficient to make proper use of the effect.

1.2.2.1.2 Enhancement of Texture Detail via Sharpening

In order to increase the apparent texture detail globally, we used a sharpening post-processing effect. Sharpening results in the enhancement of local contrast, or the enhancement of high-frequency information of the modified image.

We decided to use the well known unsharp masking algorithm (Jain 1989), with

Gaussian blur as the low-pass filter, for our implementation of the sharpening filter. A possible form of the unsharp masking expression, that we made use of, is

$$\vec{c}' := \vec{c} + i \cdot (\vec{c} - \vec{c}_{G(r)}) \tag{1.20}$$

where $\vec{c}' \in [0;1]^3$ is the resulting texel color, $\vec{c} \in [0;1]^3$ is the original texel color, $\vec{c}_{G(r)} \in [0;1]^3$ is the color of the texel at the same position in the source texture with a Gaussian blur of radius $r \in [0;+\infty)$ applied to it, and $i \in \mathbb{R}$ is the intensity of the effect.

Since Gaussian blur is a low-pass filter, the result of $(\vec{c} - \vec{c}_{G(r)})$ is a high-pass filtered image, as it is the original without the low-frequency information.

A 2-dimensional Gaussian blur may be implemented in two ways; via a 2-dimensional convolution, or, thanks to the Gaussian blur being a separable filter, using a two-pass 1-dimensional convolution. The time complexity is $\mathcal{O}(w_{kernel}^2 w_{image} h_{image})$ and $\mathcal{O}(w_{kernel} w_{image} h_{image})$, respectively.

We used the two-pass 1-dimensional method for its advantageous time complexity. Our implementation pre-computes one half of the symmetric convolution matrix on the CPU, according to the specified radius and the resolution of the texture. The execution order of separate steps of the algorithm is shown in figure 1.10.

The resulting sharpening effect (without uniform saturation and contrast enhancement from the previous section) can be seen in figure **??**.

It is noteworthy to mention one significant drawback of our approach – the resulting effect is not perceptually uniform across the whole screen. This is caused by the fact that we see the majority of the screen under an non-perpendicular angle, and so the convolution matrix is skewed from our point of view. As an improvement for the future, one might want to consider performing the Gaussian blur on the surface of a sphere centered around the HMD, to which the screen texture is projected.

Figure 1.10: Execution order of distinct components of the sharpening effect.

1.2.2.2 Tracers

"Tracers" is the colloquial name for visual tracing, the last aspect of psyechedelic-induced ASCs we implemented a replication of in our application. Tracers are shown as positive after-images of moving objects, and can appear either continuous or discontinuous (Hartman and Hollister 1963; Díaz 2010; Anderson and O'Malley 1972; Kleinman, Gillin, and Wyatt 1977). We focused on the replication of the continuous form of tracers.

The implementation required the persistence of image data between frames, which has been accomplished by the usage of an accumulation texture, one for each

screen of the HMD. The replication has been implemented via the following post-processing operation:

$$\vec{c}' \coloneqq \text{lerp}(\vec{c}, \vec{a}, \beta^{\Delta t} \cdot \alpha) \tag{1.21}$$

$$\vec{a}' \coloneqq \text{lerp}(\vec{c}, \vec{a}, \beta^{\Delta t}) \tag{1.22}$$

where

$$\text{lerp}(x, y, t) = (1 - t)x + ty \tag{1.23}$$

is the the linear interpolation function, also known as `mix` in GLSL and `lerp` in HLSL, $\vec{c} \in [0; 1]^3$ is the original color of the texel from the source texture, $\vec{c}' \in [0; 1]^3$ is the modified color of the texel stored in the output texture, $\vec{a} \in [0; 1]^3$ is the original color of the texel from the accumulation texture, $\vec{a}' \in [0; 1]^3$ is the modified color of the texel stored in the accumulation texture, $\Delta t \in [0; +\infty)$ is the duration since the previous frame, and $\alpha, \beta \in [0; 1]$ are parameters.

Another visualization of the operation is shown as an execution graph in figure 1.11.

The parameter α corresponds to the overall opacity of the effect, which is fine-tuned, so that the effect does not cause motion sickness by dominating the visual field. The parameter β controls the feedback strength, which can be understood as a parameter that influences the duration of the temporal blur or "exposition time". The Δt parameter provides resiliency to framerate fluctuations.

This is the final form of the effect that has been used in our application.

Nevertheless, the current form of the implementation has some drawbacks, that we have identified late in the development of the application.

First, we have received feedback, that the tracers should not be as noticeable during the movement or rotation of the HMD, compared to moving objects in the scene. Despite that, we decided to keep our implementation, because we find it unlikely, that the psychedelic-induced ASC would somehow change the perception of objects moving relative to some kind of absolute world frame *only*, rather than of any object that has been moved within the person's field of view.

And second, our implementation completely ignores the direction the user is looking, it only considers the rotation of the HMD. A precise eye-tracking device built-into the HMD would be required to develop a method that takes the the direction of eyes into account.

Finally, regarding discontinuous tracers, the implementation of such an effect would require keeping more temporal information than just a single accumulation texture. Probably, a 3-dimensional texture would have to be used, with the width and height of the screen and the depth of $\frac{1}{\Delta t} \cdot T$, where $\frac{1}{\Delta t}$ is the framerate, and $T \in (0; +\infty)$ is the delay of the first positive after-image of the tracers. One challenge may stem from the fact that the framerate of a VR application is typically not uniform.

Figure 1.11: Execution graph of the tracer effect. The parameter $\alpha \in [0; 1]$ is the total opacity of the effect. The parameter $\beta \in [0; 1]$ is the feedback modifier corresponding to the "duration" of the resulting blur. Finally, $\Delta t \in [0; +\infty)$ is the time since the previous frame, making the effect less influenced by framerate fluctuations. The accumulation texture is first read from, then written to.

Figure 1.12: The tracer effect is applied in the second render pass of the sharpening effect.

1.3 Complex Replication

1.3.1 Execution Order

Figure 1.13: The execution order of partial replications, making up the complex replication.

1.3.2 Experiment Automation

(a) Simplex noise from eq. 1.2.

(b) Simplex noise fBm from eq. 1.4, with coordinate offset $\vec{o} = \vec{0}$ and parameters $g = \frac{1}{l} = 1.5, k = 5$.

(c) Simplex noise fBm from eq. 1.4, with nonzero coordinate offset $\vec{o} \neq \vec{0}$ and parameters $g = \frac{1}{l} = 1.5, k = 5$.

(d) Simplex noise fBm with index offsets based on the distance to the center r, from eq. 1.7, with nonzero coordinate offset $\vec{o} \neq \vec{0}$ and pa-

Figure 1.5: The final implementation of the blended fBm Simplex noise based on the distance to the center r, from eq. 1.12, with non-zero coordinate offset $\vec{o} \neq \vec{0}$ and parameters $a = g = \frac{1}{l} = 1.5, k = 5$. Amplitude adjusted for visualization.

Figure 1.8: The first 3 components of the simplified gradient of the blended fBm simplex noise S_G, from eq. 1.18, with non-zero coordinate offset $\vec{o} \neq \vec{0}$ and parameters $a = g = \frac{1}{l} = 1.5, k = 5$. Components visualized as channels of the additive RGB color space. Amplitude adjusted for visualization and clipped to range.

Figure 1.9: The first 3 components of the simplified gradient of the blended fBm simplex noise S'_G, from eq. 1.19, with non-zero coordinate offset $\vec{o} \neq \vec{0}$ and parameters $a = g = \frac{1}{l} = 1.5, k = 5$. Components visualized as channels of the additive RGB color space. Amplitude adjusted for visualization and clipped to range.

List of Acronyms

11-ASC 11-Factor Altered States of Consciousness Questionnaire: A version of the *Altered States of Consciousness Rating Scale* psychometric questionnaire, which is based on the hypothesis that ASCs have a common core independent of the induction method which distinguishes them from the waking conscious state (Figueiredo et al. 2016; Studerus, Gamma, and Vollenweider 2010).

5-HT 5-hydroxytryptamine, also known as serotonin

5D-ASC 5-Dimensional Altered States of Consciousness Questionnaire: Like the 11-Factor Altered States of Consciousness Questionnaire (11-ASC), but with different scoring and categories (Dittrich, Lamparter, and Maurer 2010).

AI artificial intelligence

ASC altered state of consciousness: See section ?? for a complete definition and related terms.

CPU central processing unit

DCNN deep convolutional neural networks

DMT *N,N*-dimethyltryptamine: A classical hallucinogenic drug first synthesized in 1931 (Manske 1931), a psychoactive compound of Ayahuasca, the ceremonial spiritual medicine used by Amazonian natives for shamanic purposes and to bond socially in a casual setting (Mark Hay 2020).

DSP digital signal processing

EEG electroencephalograph

FOV field of view

FPS frames per second: A unit of monitor refresh rate, equivalent to hertz (Hz).

GLSL OpenGL Shading Language: A shading language used by the OpenGL graphics API.

GPU graphics processing unit: A specialized extension module providing acceleration for computer graphics computations and other parallelizable tasks.

HDRI high dynamic range imaging

HLSL High Level Shading Language: A shading language used by the DirectX graphics API.

HMD head–mounted display

LSD lysergic acid diethylamide: A classical hallucinogenic drug first synthesized in 1938 from ergotamine, an alkaloid of the ergot rye fungus (Albert Hofmann 1969).

MEQ30 30-item revised mystical experience questionnaire

MTE 'mystical-type' experience: Subjective experiences whose characteristics include a sense of connectedness, transcendence, and ineffability.

PCI Phenomenology of Consciousness Inventory: A psychometric questionnaire based on the hypothesis that different states of consciousness can be characterized in terms of phenomenological dimensions which can be quantified in terms of their intensity. The resulting pattern is assumed to be typical of a particular induction method and can be observed consistently (Figueiredo et al. 2016).

RAM random-access memory

RDG rendering dependency graph: UE4's graph-based scheduling system to perform whole-frame optimization of the render pipeline.

THC tetrahydrocannabinol: One of the psychoactive compounds in cannabis.

UE4 Unreal Engine 4: A game development engine.

VAS visual analog scale

VR virtual reality

fBm fractional Brownian motion

Bibliography

Albert Hofmann. 1969. "LSD: Completely Personal." Accessed December 6, 2013. https://web.archive.org/web/20131206032629/http://www.maps.org/newsletters/v06n3/06346hof.html.

Alessa Baker. 2021. "CustomShaderDirectoryExample: An example project for a tutorial on adding a custom shader directory." Accessed May 18, 2022. https://github.com/Sythenz/CustomShaderDirectoryExample.

Anderson, WH, and John E O'Malley. 1972. "Trifluoperazine for the trailing phenomenon." *JAMA* 220 (9): 1244–1245.

Dittrich, Adolf. 1998. "The standardized psychometric assessment of altered states of consciousness (ASCs) in humans." *Pharmacopsychiatry* 31 (S 2): 80–84.

Dittrich, Adolf, Daniel Lamparter, and Maja Maurer. 2010. "5D-ASC: Questionnaire for the assessment of altered states of consciousness." *A short introduction. Zurich, Switzerland: PSIN PLUS.*

Díaz, José Luis. 2010. "Sacred plants and visionary consciousness." *Phenomenology and the Cognitive Sciences* 9 (2): 159–170.

Figueiredo, Renato Garita, Hendrik Berkemeyer, Katharina Dworatzyk, and Timo T Schmidt. 2016. "Building a unifying database to enable flexible meta-analyses of data on altered states of consciousness."

Fischer, Rico, RM Hill, and Diana Warshay. 1969. "Effects of the psychodysleptic drug psilocybin on visual perception. Changes in brightness preference." *Experientia* 25 (2): 166–169.

Fischer, Roland, Richard Hill, Karen Thatcher, and James Scheib. 1970. "Psilocybin-induced contraction of nearby visual space." *Agents and actions* 1 (4): 190–197.

Hartman, Alan M, and Leo E Hollister. 1963. "Effect of mescaline, lysergic acid diethylamide and psilocybin on color perception." *Psychopharmacologia* 4 (6): 441–451.

Hill, Richard M, and Roland Fischer. 1973. "Induction and extinction of psilocybin induced transformations of visual space." *Pharmacopsychiatry* 6 (04): 258–263.

Hill, RM, Roland Fischer, and Diana Warshay. 1969. "Effects of excitatory and tranquilizing drugs on visual perception. Spatial distortion thresholds." *Experientia* 25 (2): 171–172.

Jain, Anil K. 1989. *Fundamentals of digital image processing*. Prentice-Hall, Inc.

Kleinman, Joel Edward, John Christian Gillin, and Richard Jed Wyatt. 1977. "A comparison of the phenomenology of hallucinogens and schizophrenia from some autobiographical accounts." *Schizophrenia Bulletin* 3 (4): 560.

Klüver, H. 1966. *Mescal and mechanisms of hallucinations Chicago.*

Klüver, Heinrich. 1942. "Mechanisms of hallucinations."

Kometer, Michael, and Franz X Vollenweider. 2016. "Serotonergic hallucinogen-induced visual perceptual alterations." *Behavioral neurobiology of psychedelic drugs:* 257–282.

Madary, Michael, and Thomas K Metzinger. 2016. "Real virtuality: a code of ethical conduct. Recommendations for good scientific practice and the consumers of VR-technology." *Frontiers in Robotics and AI* 3:3.

Manske, Richard HF. 1931. "A synthesis of the methyltryptamines and some derivatives." *Canadian Journal of Research* 5 (5): 592–600.

Mark Hay. 2020. "The Colonization of the Ayahuasca Experience." Accessed February 7, 2022. https://web.archive.org/web/20220207102112/https://daily.jstor.org/the-colonization-of-the-ayahuasca-experience/.

Olano, Mark, John C Hart, Wolfgang Heidrich, Bill Mark, and Ken Perlin. 2002. "Real-time shading languages." SIGGRAPH.

Ossi Luoto. 2021. "SceneViewExtTest: Unreal Engine 4.26 test project and plugin for SceneViewExtension and how to hook custom shaders to RDG/GraphBuilder." Accessed May 19, 2022. https://github.com/A57R4L/SceneViewExtTest.

Perlin, Ken. 1985. "An image synthesizer." *ACM Siggraph Computer Graphics* 19 (3): 287–296.

Preller, Katrin H, and Franz X Vollenweider. 2016. "Phenomenology, structure, and dynamic of psychedelic states." *Behavioral neurobiology of psychedelic drugs:* 221–256.

PsychonautWiki. 2020. *Acuity enhancement — PsychonautWiki, The Open Encyclopedia of Psychonautics.* Accessed May 19, 2022. https://web.archive.org/web/20220519164532/https://psychonautwiki.org/wiki/Acuity_enhancement.

Siegel, Ronald K, and Murray E Jarvik. 1975. "Drug-induced hallucinations in animals and man." *Hallucinations: Behavior, experience and theory* 81:161.

Studerus, Erich, Alex Gamma, and Franz X Vollenweider. 2010. "Psychometric evaluation of the altered states of consciousness rating scale (OAV)." *PloS one* 5 (8): e12412.