

VR酔い調査班 活動報告書

安田 直樹 *

海木 一佳 †

赤時 圭祐 ‡

武井 智也 §

青木 宏祐 ¶

川端 諒 ||

安田 弘樹 **

広田 大地 ††

2017年2月7日

*情報理工学部 メディア情報学科 2 回生

†情報理工学部 情報システム学科 2 回生

‡情報理工学部 情報システム学科 2 回生

§情報理工学部 情報システム学科 2 回生

¶情報理工学部 情報システム学科 3 回生

|| 理工学部 数理科学科 2 回生

**情報理工学部 1 回生

††情報理工学部 1 回生

目次

1	はじめに	4
2	ベストプラクティス	4
2.1	シミュレータ酔い	4
2.2	ユーザインタフェース	4
2.2.1	ヘッドアップディスプレイ (HUD)	4
2.2.2	アバター	5
2.2.3	武器とツール	5
2.2.4	ユーザ入力とナビゲーション	5
2.3	立体視	5
2.3.1	人がものを見る仕組み	6
2.3.2	単眼性奥行き手掛かり	6
2.3.3	oculus 内での快適な表示距離	7
2.3.4	カメラ間距離による効果	7
2.3.5	二つの画像の統合における潜在的な問題	7
2.3.6	視野角とスケール	7
3	CG	8
3.1	レンダリングパイプライン	8
3.1.1	レンダリング・2つの方向	8
3.1.2	パイプライン処理	8
3.1.3	グラフィック処理のパイプライン	9
3.1.4	アプリケーションのステージ	10
3.1.5	ジオメトリ処理ステージ	10
3.1.6	ラスターライズ処理ステージ	11
3.1.7	フラグメント処理ステージ	11
3.2	光源	11
3.3	陰影	11
3.4	テクスチャマッピング	12
3.4.1	テクスチャの種類	12
3.4.2	テクスチャの拡大, 縮小	13
3.4.3	エイリアシング	15
3.4.4	アンチエイリアシング	15
3.4.5	ミップマップ	16
3.4.6	バーテックスシェーダ	17
3.4.7	マッピングの種類	17
3.4.8	ソリッドテクスチャ	19
3.5	クォータニオン	19
3.5.1	オイラー角	19
4	立体音響	21
4.1	再現	21
4.2	方式	21
4.3	頭部伝達関数	22

4.4	トランスオーラル処理	23
4.5	分類	24
4.6	問題点	24
5	制作に用いた技術	24
5.1	terrain	25
5.1.1	概要	25
5.1.2	使用するメリット	25
5.2	破壊モーション	25
5.3	視線	25
6	SSQ	26
6.1	映像酔いについて	26
6.2	SSQ	26
7	VRの類似技術	26
7.1	VR技術	26
7.2	AR技術	27
7.3	DR技術	27
7.4	SR技術	27
7.5	MR技術	27
8	バーチャルリアリティ全般	28
8.1	バーチャルリアリティの3大要素	28
8.1.1	3次元の空間性	28
8.1.2	実時間の相互作用性	29
8.1.3	自己投射性	29
8.2	バーチャルリアリティの認知	29
8.3	バーチャルリアリティの用途	29
8.3.1	3C-Creation	29
8.3.2	3C-Control	30
8.3.3	3C-Communication	30
8.3.4	3E-Elucidation	30
8.3.5	3E-Education	30
8.3.6	3E-Entertainment	30
9	まとめ, 今後の展望	30

1 はじめに

本プロジェクトでは、「VR 酔い」についての調査を行うことを目的とする。2016 年に、Oculus Rift CV1 や、PlayStation VR が発売され、バーチャルリアリティ(VR)への注目が高まった。しかし、VR コンテンツを容易に作成できるようになるにつれて、VR の危険性が軽視されつつある。VR の危険性の代表的なものとして「VR 酔い」が存在する。本プロジェクトでは、VR コンテンツが、従来の 3D ゲームなどのコンテンツとどのように違うかを理解する。また VR 酔いを起こさない、低減するコンテンツを作成するにはどのようにすればいいかを調べ、実際に作成された VR コンテンツが VR 酔いを起こし辛いものであるかを評価する。その為に以下の事を調べた。

- ベストプラクティス
- CG
- 立体音響
- SSQ
- VR コンテンツに含むギミックの実装, その技術

また、VR の現状を理解するために、バーチャルリアリティ全般、及び AR,MR といった VR と類似した名称の技術についても調べた。

2 ベストプラクティス

快適な VR 体験を作るためには、基本的な設計方針に基づいて作る必要がある。不適切な VR 体験はシミュレータ酔いを引き起こし、不快感を与える。今回は、Oculus 社が提供している「ベストプラクティス (<http://static.oculus.com/documentation/pdfs/ja-jp/intro-vr/latest/bp.pdf>)」を読んで不快感を与えない VR 体験の作成手法を学んだ。

2.1 シミュレータ酔い

VR 酔いに限らず、何らかのシミュレートされた環境によって発生する不快感を「シミュレーター酔い」という。このシミュレーター酔いは主に視覚と体の感覚の相違が原因とされている。例えば、VR 内のキャラクターの移動速度を現実のそれと同様にする、処理落ちによる映像の遅延を発生しないようにする、など様々な点で注意する必要がある。また、このシミュレーター酔いは乗り物酔いと類似している。乗り物酔いは、普段は体験しない揺れによって引き起こされるがシミュレーター酔いは、ユーザーは移動していないが、ユーザーが見る視覚情報が移動していると感じさせることによって発生する。そして、乗り物酔いを起こしやすい人は、シミュレーター酔いも起こしやすいという調査結果もある。本プロジェクトでは、VR 酔いを調べることが目的だが、そのためにシミュレーターによる不快感を評価する SSQ(Simulator Sickness Questionnaire) と呼ばれるテストを用いる。この SSQ の詳細については別項で記述する。SSQ を用いることで、シミュレートされた体験がどの程度不快であるかを測ることができる。

2.2 ユーザインタフェース

2.2.1 ヘッドアップディスプレイ (HUD)

ヘッドアップディスプレイ (以下 HUD) とはゲーム画面に地図、点数、プレイヤー情報などの情報を手前に表示させるものである。Oculus では通常 HUD を推奨していない。なぜなら HUD が画面内のすべて

のオブジェクトの手前に表示されることにより、立体視のゲームでは遠近感に矛盾が生じたり、各目の映像の統合を困難にしたりしてしまうためである。HUDを使わない解決策としては、HUDで表示させたい情報を画面内の環境そのものに統合する方法がある。例を挙げると、HUDで表示していたミニマップやコンパスを代わりにプレイヤーの身に着けているもの、建物内にある実際のマップに目を落として確認するなどである。

2.2.2 アバター

アバターとは、バーチャル世界におけるユーザの体の視覚的な表現で、ユーザの位置、動作のジェスチャーに対応している。アバターの長所はユーザにバーチャル世界における縮尺や体の大きさを強く認識させることができる点。アバターの短所はアバターの提示することによりユーザの自己受容感覚（座ってゲームをしているのにゲーム内では歩行している状態など）に矛盾が起こり、奇妙に感じる可能性がある点である。

2.2.3 武器とツール

FPS (First Person shooting) では、武器は通常、画面の下のほうに表示され、ユーザが持っており照準を合わせているように配置されている。この方法を立体視で実装する場合、複雑になる。武器やツールのレンダリングをカメラの直前にすることで武器とほかのシーンとの間を見る際に、ユーザは眼の輻輳 (両目が同時に内側を向く目の動き) を大きく変化させる必要がある。さらに、武器やツールが視点にあまりに近いために左右のビューが著しく異なり、3次元ビューに解像することが難しくなる。

2.2.4 ユーザ入力とナビゲーション

Oculus Rift を装着するとキーボード、マウス、ゲームパッド、モニターのいずれもユーザがVRに入るとみることができないことを把握する。いったんVRに入ると、これらのデバイスは触覚だけで操作することになる。特にキーボードの使用は面倒になる。理想的なものとは言えないが現状の入力デバイスではゲームパッドを使うことが最善の選択である。ユーザは両手でゲームパッドを握ることができ、慣れれば慣れるほど、視覚的な頼り無しで利用する際の安心感が大きくなる。

2.3 立体視

VR技術によって映像を見せるとき、立体的でなければならない場面も数多く存在する。そこで、人はどのようにして立体の奥行きを認識しているのか、そもそも人がものを目で見る仕組みとはどのようなものなのかを述べていく。

2.3.1 人がものを見る仕組み

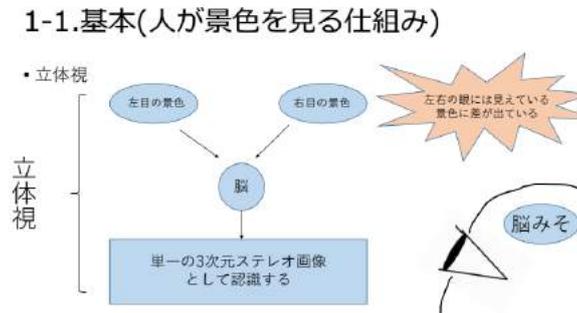


図 1: 人がものを見る仕組み

左目で見る景色と右目で見る景色はそれぞれ違いが存在する。その違いを脳が統合し、単一の3次元ステレオ画像として認識している。この流れを立体視といい、これこそが人がものを立体的にとらえることができる理由である。

2.3.2 単眼性奥行き手掛かり

人が奥行きを認識できるのは片方の目それぞれのもつ働きによるところが大きい。以下、用語が多く登場するためここでまとめて定義しておく。

IPD 瞳孔間距離。52～78mm 程度。平均は 63.5mm

IAD oculus rift のレンズ中央間の距離。63.5mm

ICD バーチャル空間を映す二つのレンダリングカメラ間の距離。
適切な距離は場合による。

重要な奥行き手掛かりは先ほど述べた立体視以外にも存在する。以下ではそれらを紹介する。

曲面透視 複数の直線は1点に収束するように伸びている

相対スケール オブジェクトは遠いと小さく見える

遮蔽 近くのオブジェクトは(重なっていれば)遠く、オブジェクトの一部を隠す

空気遠近法 遠くのオブジェクトは霞む(空気の光屈折のせい)

テクスチャ識別 反復パターンを遠目で見ると、ぎっしり詰まって見える

照明 ハイライトとシャドウでオブジェクトの形と位置の認識を助ける

2.3.3 oculus 内での快適な表示距離

VR 空間内でオブジェクトに注視するとき目の快適さにかかわる要素が二つある。

調節デマンド 奥行き面に焦点を合わせるために (調節), 目のレンズの形状をどのように変化させる必要があるのかの度合い

輻輳デマンド 両目の視線が特定の奥行き面で交わるように, 目をどのくらい内側に向ける (寄り目にする) 必要があるかの度合い

VR 空間内では調節デマンドは固定で輻輳デマンドは可変である。これは現実世界とは異なる特殊な状況である。その現実世界との差異を埋めるため, VR 空間内ではレンダリングに工夫を加える必要がある。1.3m 離れた画面を見ているようにレンダリングすることが重要で, メニュー画面や, 重要なアイテムなどは 0.75m ~ 3.5m の範囲でレンダリングすることが重要。¹

2.3.4 カメラ間距離による効果

ICD(カメラ間距離)を変化させると輻輳デマンドが変化したり, VR 空間内での自分自身のサイズ感が変化する。そのほかにも, 次のような効果が得られる。

広くする 奥行きが誇張される (ハイパーステレオ)

狭くする 奥行きが平板になる (ハイポステレオ)

ICD=IPD の場合, 真に迫ったスケールと奥行きを VR 空間内で実現できる。ICD を変化させることは構わないが, その距離は IPD に比例していなければならない。

2.3.5 二つの画像の統合における潜在的な問題

左右の目に映る風景がかなり異なる状況は現実世界でいくらかでも存在する。例えば, 視力検査の時を思い起こしてみると, 片方の目は数メートル先にある電光掲示板を見ている一方でもう片方は暗闇を見ている。VR 空間内でこのようなことが生じる場合, 少し気を付けなければならない。レンダリング効果は両方の目に正しい視差で映る必要があり, 2 つの目の間の画像が異なっていないことを確認することが重要である。

2.3.6 視野角とスケール

バーチャルカメラの FOV について述べる。FOV とは視野角のことである。まず, 用語の定義を述べる。

dFOV ユーザーの物理視野角のうち, VR コンテンツが占める部分 (ハードウェアや光学系の物理的な特性)

cFOV 任意のタイミングでバーチャル世界の中でカメラがレンダリングする範囲

基本的に開発時はデフォルトに合わせていればいいが, 注意しなければならない最も重要なポイントは, **cFOV** と **dFOV** は完全に一致しなければならないということである。

$\frac{dFOV}{cFOV}$ をスケールと呼び, これが常に 1.0 でないと VR 酔いにつながる。

$\frac{dFOV}{cFOV} \neq 1$ の時, レンダリングされたシーンが歪曲する。スケールが 1 でないまま cFOV を調整したとき,

¹VR 内のすべてのオブジェクトに焦点が合っていると, 不快感を訴える者がいるらしい (未確認)
また, ユーザーが見ている対象が把握されている状況では, 被写界深度エフェクトを用いると, 没入感と快適さの両方を実現できる (一部の開発者談)

- シミュレータ酔いが発生
- 前庭動眼反射の不応
- oculus を脱着した後，視覚運動機能が低下

などといった影響が発生する。²

【補足】

FPS ゲームには標的を狙うためにズームをすることがよくある。しかし、これは非常にトリッキーな操作で、頭部の運動と見かけ上の世界のモーシヨンの連携が崩れて不快感を生じさせる場合がある。

3 CG

VR 体験を作成するにあたって、CG 技術の使用は不可欠である。そこで、CG 分野において用いられる技術・手法についていくつか調べた。調査には、和歌山大学「ゲームグラフィックス特論 (<http://www.wakayama-u.ac.jp/~tokoi/lecture/gg/>)」の講義資料を主に参考に行っている。大きく以下の5つについて調べた。

- レンダリングパイプライン
- 光源
- 陰影
- テクスチャマッピング
- クォータニオン

3.1 レンダリングパイプライン

レンダリングとは、「直方体」「三角錐」といった図形の情報と「青色」「木の葉状」といった材質の情報と光源情報と視覚情報を組み合わせて、画像情報を作り出すことを言う。ここでは、その手順について述べる。

3.1.1 レンダリング・2つの方向

サンプリング 処理に時間がかかるが高品質な映像生成が行える

ラスタライズ リアルさに欠ける場合があるが、リアルタイムで行える (リアルタイムレンダリング) ので VR 向き

3.1.2 パイプライン処理

パイプライン処理とは並行処理の1つで、レンダリングの処理の流れをいくつかのステージに分割する。最大で分割した個数倍だけ処理速度が速くなる。しかし、その速度は分割されたステージの中で最も処理速度が遅い部分に合わせて決まる。これをボトルネックという。

²SDK を用いることでスケールを変更せずに cFOV と dFOV を変更することが可能

パイプライン処理による速度向上

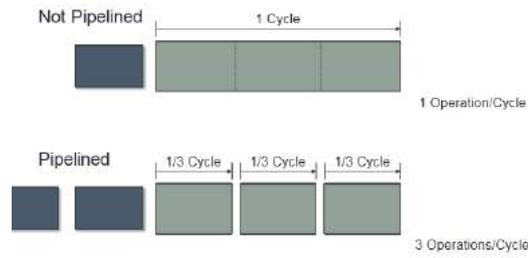


図 2: パイプラインの様子

ボトルネック

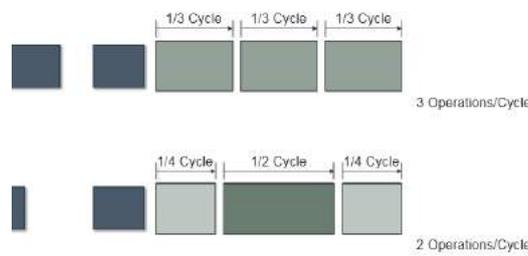


図 3: ボトルネック

3.1.3 グラフィック処理のパイプライン

実際にコンピュータがどのようにグラフィック処理をしているのかを述べる。CPUが行うのはアプリケーションでの処理とジオメトリ処理であり、GPUが行うのはジオメトリ処理とラスタライズ処理とフラグメント処理である。

グラフィックス処理のパイプライン

・ハードウェアの概念上のステージ

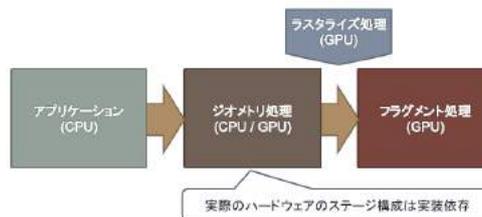


図 4: ハードウェアの概念上のステージ

3.1.4 アプリケーションのステージ

CPU 側のソフトウェアで行っていることは以下のとおりである。

- 図形の情報を次のステージに送る
 - － 形状の情報は基本図形 (レンダリングプリミティブ) で記述されている
 - － 点, 線分, 三角形など
- ジオメトリデータの出力 (このステージで出力される図形データ)
 - － 頂点の情報 (位置, 色, 法線ベクトル等) の集まり+図形の種類
- 他のソースからの入力を処理する
 - － マウス, キーボード, ジョイスティックなどからの入力
- 衝突検出
- 時間に関する処理
 - － 座標変換によるアニメーション
 - － スキニング・ジオメトリモーフィング

他のステージでは実行できないすべての種類の計算を行っている。

3.1.5 ジオメトリ処理ステージ

CPU と GPU でおこなうジオメトリ処理は頂点単位での処理や図形単位での処理が行われている。

- 頂点単位に行う処理
 - － 頂点位置の座標返還
 - － 頂点の陰影付け
- 図形単位に行う処理
 - － クリッピング (視野角の外にある部分をカットすること)

ジオメトリ処理のパイプライン



図 5: ジオメトリ処理のパイプライン

3.1.6 ラスタライズ処理ステージ

GPU側で行われるラスタライズ処理は以下のことが行われている。(ラスタライズについては上述のとおり)

- 図形データから画像データを生成する
- 頂点属性の補間をする

3.1.7 フラグメント処理ステージ

最終ステージであるここでは画素単位での処理を行う。エフェクトととらえるとわかりやすい。

- フォグ
 - － 霧の効果, 大気遠近法
- アルファテスト, ステンシルテスト
 - － 型抜き
- デプステスト
 - － 可視判定
- ブレンド処理
 - － カラーバッファ上での色の合成 (透明度の処理など)

3.2 光源

光源は直接光と反射光の2つに分かれて人間の目に届く。直接光とは光源から放射され直接目に届く光であり、反射光は光が物体表面で反射して目に届く光である。基本的な光源は3種類ありどれも光源は点である。平行光源は光源が無限の彼方にあるため、放射方向が揃っており、大きさがとても小さく(点)になっている。太陽がこれにあたる。点光源は1点から球面上にすべての方向に均等に光を放射する。電球がこれにあたる。スポットライトは特定の方向に光を強く放射する。懐中電灯がこれにあたる。これらの光源とは別に光源が面になっている面光源と呼ばれるものもある。面光源とは大きさを持つ光源で、点光源に対してやわらかい影を作る。反射光は二色性反射モデルで拡散反射光と鏡面反射光の2種類に分けられる。拡散反射光は散乱により指向性を持ってないため、すべての方向に均等に放射される。鏡面反射光は指向性を持っており、正反射方向を中心に放射される。物体表面上にハイライトを生成して物体を輝かせて見せる。

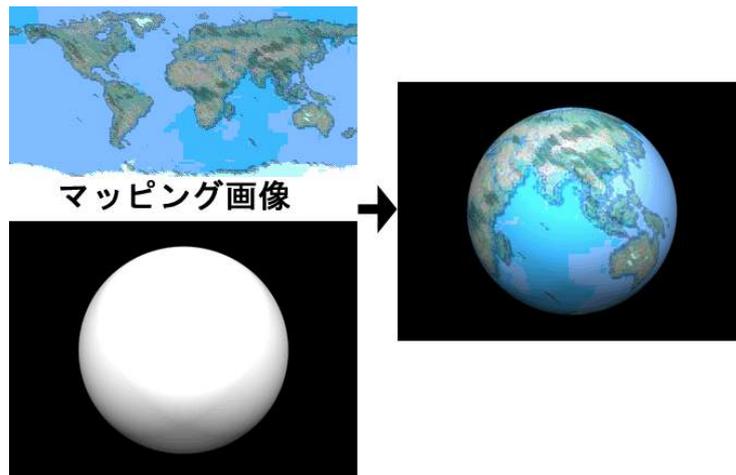
3.3 陰影

陰影とは光の当たらない、影の部分のことである。面内の陰影のつけ方にはフラットシェーディング、グーローシェーディング、フォンシェーディングの3つの方法がある。フラットシェーディングとは、陰影計算を三角形単位で行う方法である。このため、三角形上のすべての画素は同じ色になる。特徴としては、高速に処理ができ、3つの中で最も実装が簡単である。グーローシェーディングとは、陰影計算を三角形の頂点単位で行い、三角形上の各画素の色を補間により求める方法である。特徴としては、少ない三角形で滑らかな曲面を表現することが可能である。ただし、ハイライトの消失やマッハバンドの発生などの問題がある。マッハバンドとは、微妙に濃淡の異なるグレーの領域が接触している場合に、暗いほうの領域の境界付近は

より暗く，明るいほうの領域の境界付近はより明るく強調されて見える錯覚現象．フォンシェーディングとは，陰影計算を画素単位で行う方法である．特徴としては3つの中で一番細かい部分まで陰影を表現できるが，頂点単位の陰影計算に比べて，計算コストが高い．

3.4 テクスチャマッピング

テクスチャマッピングとは，物体の表面に画像を貼り付ける処理のことである．物体の形状を表すオブジェクトをテクスチャと呼び，その物体（テクスチャ）の表面に貼り付ける画像をマッピング画像と呼ぶ．テクスチャマッピングの処理を行うことで VR の映像で使用する一つ一つの物体を形成することが出来る．



〈図〉 テクスチャマッピング

図 6: テクスチャマッピング

3.4.1 テクスチャの種類

テクスチャマッピングにはいくつかの種類がある．今回調べたのは，ハイライトマッピング，バンプマッピングである．ハイライトマッピングは，物質に光を与える処理である．元の画像に，陰影を与えるような画像をマッピングすると，元の画像に陰影が付与された画像を作り出すことが出来る．バンプマッピングは，物質の表面に材質のような見た目を与える処理である．たとえば球状の物質があったとき，その球の表面に細かい模様を与えることによって，つやのある感じやざらざらした感じを視覚的に表現することが出来る．さらにこの二つのマッピングを組み合わせたハイライトバンプマッピング処理も存在する．

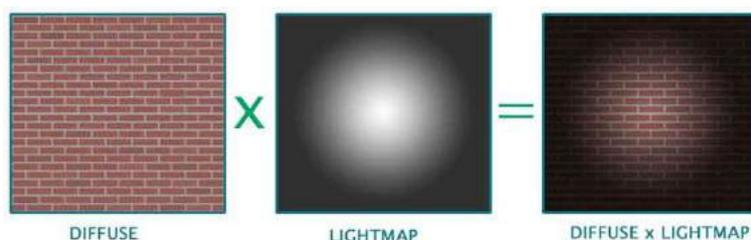


図 7: ハイライトマッピング

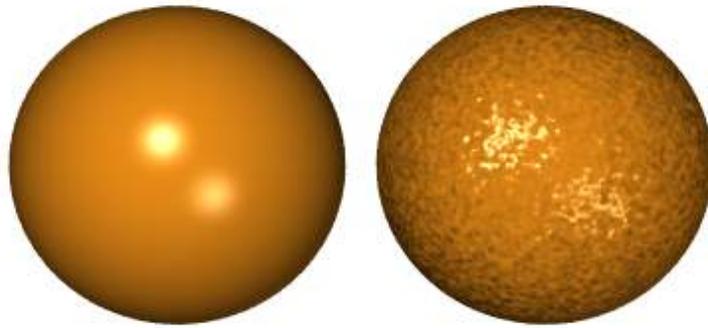


図 8: バンプマッピング

3.4.2 テクスチャの拡大, 縮小

テクスチャを拡大, 縮小する方法は大きく分けて, ニアレストネイバー法, バイリニア法, バイキュービック法の三つがある. ニアレストネイバー法は, 特に補正等をかけずにそのまま拡大, 縮小する方法である. 元の画像サイズよりも大きくすると, ドットのギザギザが目立ち, 反対に縮小するとカリカリした画像になってしまい, 画質が低下してしまう. そこで, 拡大, 縮小する際に周辺の画素からの距離を加重平均した処理を行いドットのギザギザを目立たなくする手法がバイリニア法である. バイリニア法では周辺 4 画素を加重平均して拡大, 縮小された画像を生成する. 周辺の画素を加味することでドットのドットの間画素の差が減り, なめらかな画像になるが, 少しぼやけるような画像になってしまう. バイリニア法では, 周辺 4 画素の加重平均を取るが, これを 16 画素の加重平均にしたものがバイキュービック法である. 先ほどのバイリニア法での弱点であったぼやけるような印象がバイキュービック法では解消され, きれいに拡大, 縮小を行うことが出来る.



図 9: ニアレストネイバー法



図 10: バイリニア法



図 11: バイキュービック法

3.4.3 エイリアシング

テクスチャを縮小するとエイリアシングと呼ばれる現象が発生する。エイリアシングとは、画像を縮小したときに画素の色がその領域の平均色にならない現象のことである。画像を縮小すると周辺の複数の画素が一つの画素としてまとめられることになる。まとめられる画素が似たような色であればエイリアシングは発生しにくいですが、まったく異なる色の画素が一つの画素として標本化されるとエイリアシングが発生しやすくなる。エイリアシングが発生することによって画素にノイズが発生し、縮小後、正しい色を識別できなくなってしまう現象が発生する。

3.4.4 アンチエイリアシング

エイリアシングを解消する手法としてアンチエイリアシングがある。アンチエイリアシングとは、画素のギザギザをなめらかにする手法である。画面に対して斜めに描かれた線や、緩やかな曲線などは、ドットで表現したとき、画素が粗いと線のギザギザが目立つという現象が起こる。ギザギザになった箇所に対して、その周辺の色を平均色を取って線の境界の差を埋めることで、ギザギザを目立たなくすることが出来る。アンチエイリアシングをすることで、ある点の周辺の色が極端な変化が少なくなるため、テクスチャを縮小するときの平均色を取る処理でエイリアシングが発生しにくくなる。

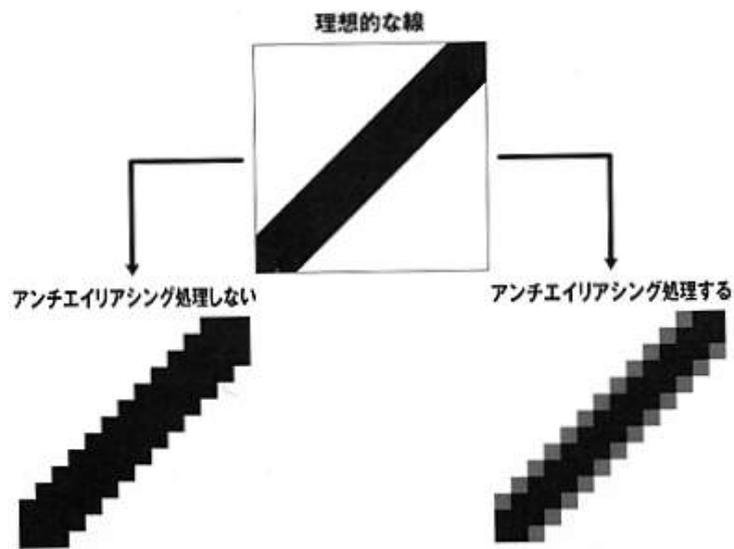
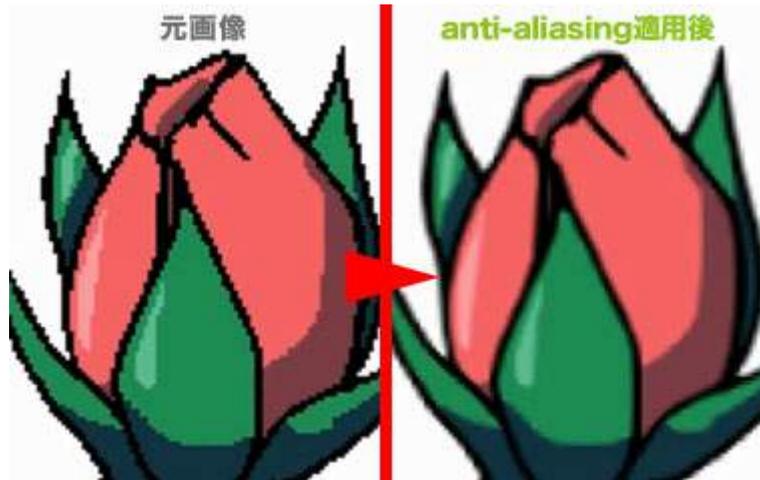


図 11-3 エイリアシング防止(その2)

図 13: アンチエイリアシング

3.4.5 ミップマップ

エイリアシングと同様に、画像を縮小するときの画像処理としてミップマップというものがある。ミップマップとは、画像を縮小したときに発生するカリカリ感をなくしてなめらかにする手法である。縦や横、斜めに複数の線が入っていたり、縞模様のような画像を縮小すると線と線が重なりあって特殊な模様が出来たり、カリカリした画像になってしまう。線が重なって通常とは異なって見える箇所を少しぼかすことによって、自然な見え方にする処理がミップマップである。

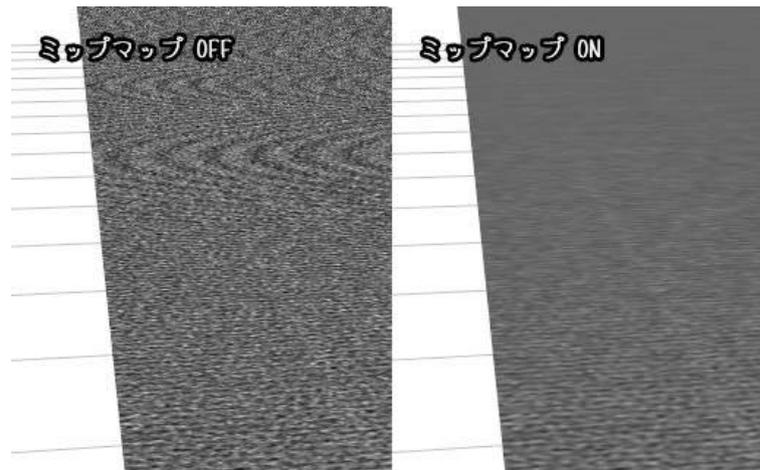


図 14: ミップマップ

3.4.6 パーテックスシェーダ

静止画ではなく、映像として、画像を動かす処理として、パーテックスシェーダという手法がある。パーテックスシェーダは、頂点の座標だけを計算して物質を変形させる処理である。ゲームなどでは、ファーストパーソン（プレイヤー）が動くことによって、動いた先にある物質が大きく見えたり、反対に動いた方向とは逆向きにある物質は小さくなって見える。また、ゲーム内のカメラの向きを変えることによって物質の形が変化したりする。このように物質に形に変化を加える処理では、その物質の各頂点だけに注目して、その頂点の座標を変更して物質全体を移動もしくは変形することが出来る。たとえば三角形を変形させる際には、三角形の三つの頂点の座標だけを変更して、各頂点を線で結ぶことで、簡単に変形させることが出来る。

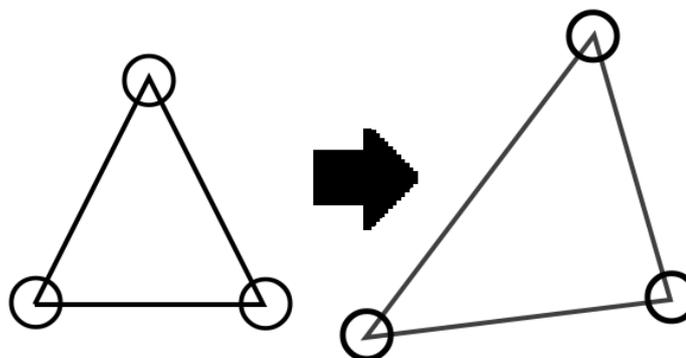


図 15: パーテックスシェーダ

3.4.7 マッピングの種類

冒頭で説明した通り、テクスチャマッピングは、テクスチャに画像を貼り付ける処理であるが、マッピングにはさまざまな種類がある。今回調べたのは、球面マッピング、円柱マッピング、平行マッピング、UVマッピングの4種類である。球面マッピングと円柱マッピングはその名の通り、それぞれ球面のテクスチャ、円柱のテクスチャに画像をマッピングする処理である。球面マッピングの例として、球状の物質に、世界地

図の画像をマッピングすることで地球儀を表現するなどがある。円柱マッピングの例として、円柱の物質に、商品のラベルの画像をマッピングすることでアルミ缶やスチール缶を表現するなどがある。平行マッピングは、立方体や直方体に画像をマッピングする処理である。立方体や直方体は面がすべて平面的なので、二次元のマッピングと似たようなマッピング処理となる。UV マッピングは、3D 空間の物質（図形）とテクスチャを組み合わせるマッピング処理である。人間や動物などの複雑なオブジェクトに対して、部品ごとにマッピングを行っている。



図 16: 球面マッピング



図 17: 円柱マッピング

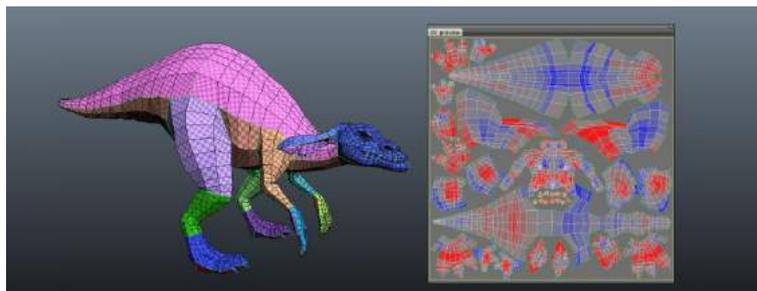
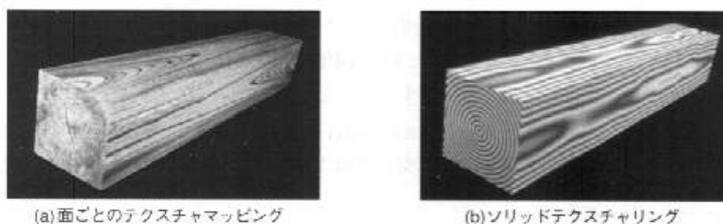


図 18: UV マッピング

3.4.8 ソリッドテクスチャ

最後に、特殊なマッピングとしてソリッドテクスチャ（3D テクスチャ）について調べた。ソリッドテクスチャは、3D 空間の物体に三次元のテクスチャをマッピングするものである。たとえば立方体や直方体にマッピングすることを考えたとき、面ごとにマッピングを行うと、面と面の境界が不連続になってしまう。ソリッドテクスチャを用いることで、3D としての物質全体をマッピングするので、境界が不連続にならずにマッピングすることが出来る。



◆図5.55 テクスチャマッピングとソリッドテクスチャリング

図 19: ソリッドテクスチャ

3.5 クォータニオン

クォータニオンとは、3DCGなどで「物体の姿勢」を表す為に用いられる。クォータニオンは日本語で「四元数」と書き、名前の通りに4つの次元を持つ数である。クォータニオンを使うと「物体の姿勢」を表すことができると書いたが、物体の姿勢を決定するには回転を扱う必要がある。2次元平面における座標の回転は複素数平面を用いると表現できる。これは1つの実数と1つの虚数を用いる。対してクォータニオンでは1つの実数と3つの虚数を用いて回転を行う。クォータニオンは以下のように表記する。

$Q = (t; x, y, z)$ t:実部

x,y,z:虚部

となっている。

3.5.1 オイラー角

このクォータニオンと別に、3次元の物体の姿勢を表す方法として「オイラー角」と呼ばれる手法が存在する。オイラー角は、3つの回転角を与えて、3軸を指定した順序で回転させるものである。

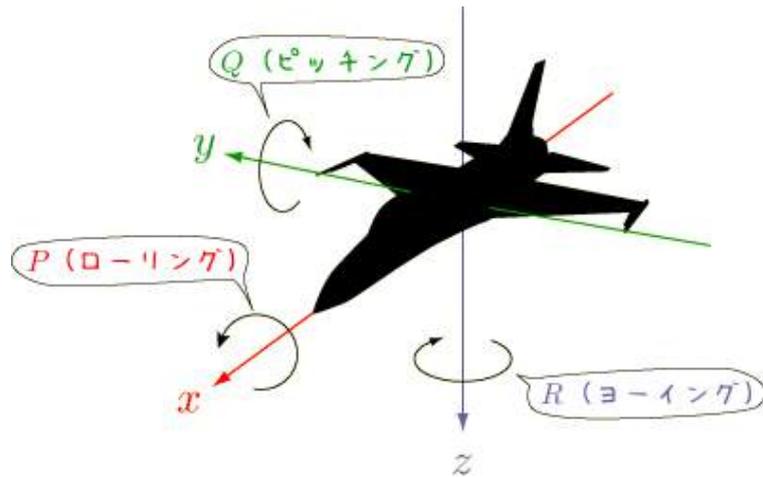


図 20: オイラー角による回転のイメージ図

このオイラー角は物体をどのように回転させるかが直感的でわかりやすいという利点があるが、欠点も存在する。その欠点に、「ジンバルロック」と呼ばれる現象が存在する。順に回転する3軸の内、2番目に回転する軸を90度回転すると、第1、第3の回転軸が同一になり回転の自由度が失われてしまう。以下のサイトでジンバルロックのシミュレートを行うことができる。

<http://wonderfl.net/c/uTHS> ジンバルロックを体験してみよう！2 (PV3D ジャイロスコープ編)

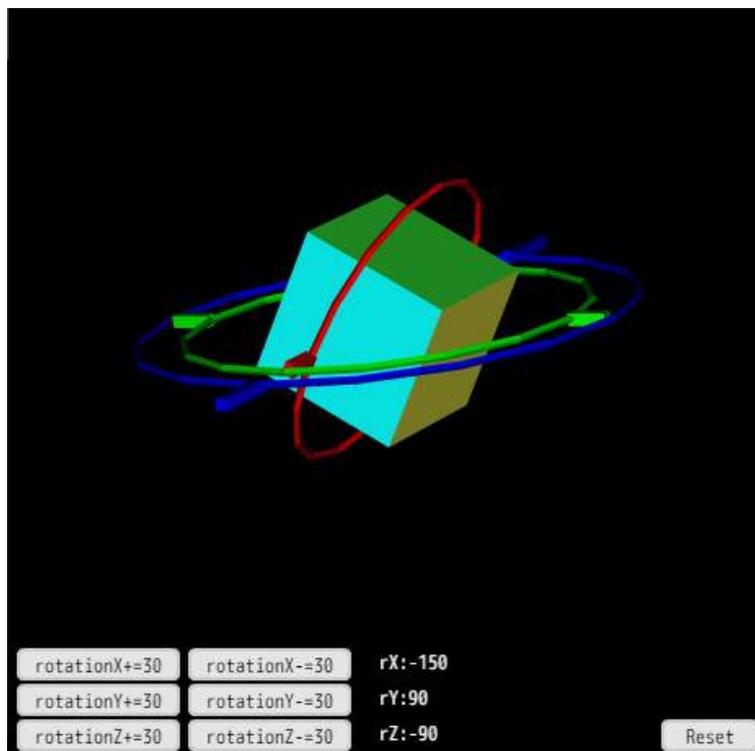


図 21: ジンバルロックの発生例

クォータニオンを用いた場合、ジンバルロックが発生しないという利点が挙げられる。

4 立体音響

3D バトルのような VR ゲームを考えたとき、敵が左右前後どこから近づいてくるかという情報は重要である。また、3D バトルに限らず VR ゲームを作ることに於いて、画面上のどの方向から音が発生するかを表現するのはゲームの臨場感を作り出す意味でも重要な要素であると考えられる。立体音響は、そのような、“どこから音が流れているか”をリアルに表現できる手段の一つである。立体音響は VR でも重要な役割を果たしていると考えたので、立体音響について調べることにした。

4.1 再現

立体音響を再現するには、音量差の利用、時間差の利用、周波数特性の変化の利用、位相の変化の利用、残響の変化の利用の主に 5 種類がある。音量差は、距離による音量の減衰により音源の音像定位を再現する。たとえば車が遠くから自分の近くにやってくる時、遠くにいる間は音は小さく聞こえ、だんだんと近づいてくると音が大きくなる。これを再現することで、車が近づいてきていることを音だけで表現することが出来る。時間差は、音波が達成する時間差により音源の音像定位を再現する。二つの同じ音源を用意して、同じ音楽を再生するとき、一つの音源で再生開始したあと、もう一つの音源をほんの少しだけ遅らせて再生すると音が二重に重なって聞こえる現象がある。これを利用して、音が立体的に聞こえるような音源を作り出すことが出来る。周波数特性の変化は、音波の伝達や遮蔽による周波数の変化により音源の音像定位を再現する。位相の変化は、音波の伝達や遮蔽による位相の変化により音源の音像定位を再現する。残響の変化は、残響特性（インパルス応答）により周辺環境の音場を再現する。

4.2 方式

音源の方式には、ステレオ方式とバイノーラル方式の 2 種類がある。ステレオ方式は、複数のマイクで収録し、スピーカーから別個に出力する方式である。複数のマイクで収録する分、さまざまな空間から音を拾うことが出来るので立体的に聞くことが出来る。バイノーラル方式は、両耳の鼓膜に届くべき信号を直接出力する方式である。人間の耳に聞こえる音に合わせた音源であるため、ステレオ方式よりもリアルに音を聞くことが出来る。この音源には、耳や頭、肩などで反射、回折される音が反映されている。バイノーラル方式の音源を収録するためには、ダミーヘッドと呼ばれるものが利用される。人間の頭部の形をしたものを用意し、耳の部分にマイクを設置し、耳や頭で反射する音を拾い、人間が聞く音に近い音を収録できるように作られている。



図 22: ダミーヘッド

4.3 頭部伝達関数

人間が聞く音を数値的に解析する手法として、頭部伝達関数と呼ばれるものがある。頭部伝達関数は、耳殻、人頭、肩まで含めた周辺物によって生じる音の変化を伝達関数としてプロットしたものである。スピーカーから発する音が、人間の耳に到達するまでには、その間にある障害物や壁に反射することによって、元の音源とは少し異なっている。頭部伝達関数は、それらの情報も含めた音源の周波数や大きさをグラフ化して表現することが出来る。

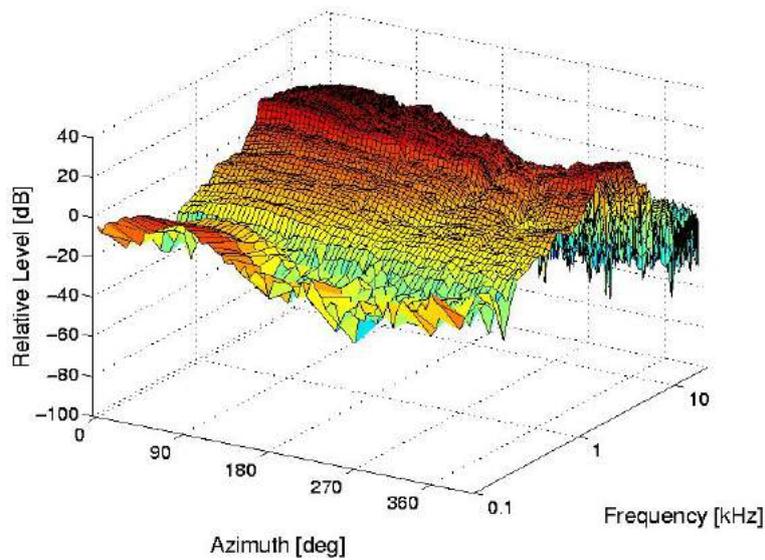


図 23: 頭部伝達関数

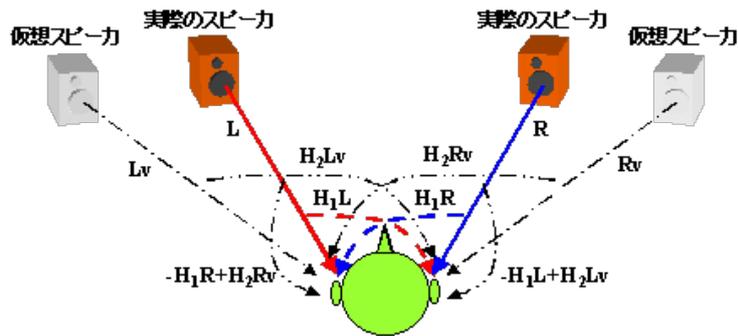


図 24: 頭部伝達関数

4.4 トランスオーラル処理

バイノーラル方式で収録された音源を聞く際にはヘッドフォンを使用するが、ヘッドフォンを使用せずにステレオスピーカーで聞くためのシステムをトランスオーラルシステムと呼ぶ。スピーカーを二台用意し、片方を左耳に入ってくる音源、もう片方を右耳に入ってくる音源として再生することでステレオ音源として聞くことができる。ところがヘッドフォンを使用していないため、左耳で聞くためのスピーカーから発生する音が右耳でも聞こえ、同様に、右耳で聞くためのスピーカーから発生する音が左耳でも聞こえてしまう、クロストークと呼ばれる問題が生じる。これを解消するために使用される手法がトランスオーラル処理である。スピーカー間のクロストークを計算し、各耳に聞こえるべき音のみが聞こえるようにキャンセリング信号を重畳（ちょうじょう：繰り返し畳み掛ける）することでクロストークを解消することができる。



図 25: トランスオーラル処理

4.5 分類

立体音響を作り出す方法を分類すると、静的か動的、録音か生成かの計 4 種類に分けられる。静的録音では、複数のマイクで録音する、もっともスタンダードな収録方法である。静的生成では、立体音響レンダリングと呼ばれる技術を用いる。動的録音では、ステージ音響シミュレーションやコンサートホールのリアルタイムシミュレーションをして音源を録音する。動的生成では、対話型の音波伝達計算を行い音源を生成する。

4.6 問題点

立体音響にはいくつかの問題点があり、今回は二つの問題点を調べた。一つは、立体音響を聞く人が激しく移動すると音を正確に表現できないことである。立体音響はその場の音場を立体的に表現する手法であるので、聞く人が移動すると、耳に聞こえてくる音波にずれが生じ、立体音響として聞こえないという問題が生じる。もう一つは、音響を作り出すのに多くのスピーカーを要することである。立体音響はあらゆる場所からの音を収録してそれらの音源を組み合わせることで立体的な音を生み出しているため、多くのスピーカーが必要となる。多くのスピーカーを要するため、それらをすべて設置できるだけの部屋を用意する必要があり、またそれだけの環境を用意するだけの金銭的なコストがかかるという問題がある。

5 制作に用いた技術

ここでは、今回作成した VR コンテンツに含まれるギミック、技術についての説明をする。以下の 3 つについて調べた。

- terrain
- 破壊モーション
- 視線制御

5.1 terrain

5.1.1 概要

Terrain とは、Unity 内で使用可能な 3D オブジェクトの種類である。名前に通りゲーム内で使用する地形として用いる事が出来るオブジェクトであり、他の 3D オブジェクトには無い様々な操作を行う事が可能。幾つかの例を挙げると、表面へ自由にテクスチャを塗る、山や谷のような起伏を付ける、木や草のようなオブジェクトを群生させる、といったような事が行える。これらの機能を組み合わせるだけでも、ゲーム内で使用する地形が手軽に作製できる。

5.1.2 使用するメリット

シーン内に Terrain を設置して基盤にすると、他のオブジェクトをドラッグ操作で設置する際に設置先の Y 座標が自動的に Terrain の起伏に合わされるため、オブジェクト配置が行いやすくなる。また、複数のテクスチャを別々の箇所へ同時に塗れるという点も、他の 3D オブジェクトにはない長所である。今回の制作では、地形の起伏を用いる事でプレイヤーキャラが移動出来る範囲をプレイヤーにも分かりやすく視認できるようにした。また、テクスチャの塗りやすさを活用する事によって、森林、遺跡、その二つを繋ぐ草原といった三か所のエリアを一枚の Terrain 内に分割して収める事が出来た。

5.2 破壊モーション

我々の班では、道を塞ぐ岩を破壊するというギミックをゲームに取り入れることにした。当初は、岩の破壊を表現する方法として岩のオブジェクトに切れ目を入れることにより、衝撃が加わった際に小さいオブジェクトになって飛散させる方法と、衝突後に煙のエフェクトを発生させ、岩が消滅する瞬間をごまかすという 2 つの方法を考えていた。前者は破壊された際の表現が自然な感じを出せるのが良く、可能であれば採用したい方法であった。しかし、開発スケジュールにあまり余裕がなかったことや、同様の表現が可能になるアセットが有料のものしかなかったため、この方法は断念することとなった。後者は前者と比べて破壊モーションの表現の自然さが劣るものの、実装が簡単であることから、採用となった。ゲーム内では、道を塞いでいる岩が、転がり落ちてくる岩によって破壊されるという形をとることとした。また、破壊音を使用することにより、岩破壊時の臨場感を補うようにした。スクリプトでは、衝突した瞬間に煙のエフェクト・破壊音の再生を有効にし、WaitForSeconds を使用して 2 秒ほど待機した後に Stop にて煙のエフェクトを無効化・Destroy で道を塞ぐオブジェクトの破壊を行うようにした。

5.3 視線

Oculus などのヘッドマウントディスプレイを用いて VR を行うと、直接コントローラを視認することができなくなってしまうため、操作性はあまり良くない。また、コントローラを用いた操作は仮想世界と現実世界の齟齬をきたし、VR 酔いや VR 空間への没入間を下げる可能性がある。そのため、VR において視線を用いた操作の優位性は非常に高い。よって、我々も視線を用いた操作を一部導入した。本来ならば、UI などにも導入し、できるだけ VR らしい操作感を実現したかったが、期間的問題から一部のみとなった。目

線による操作の実装においてはカメラからレイを飛ばすことで実現させた。カメラからレイを飛ばし、「何に当たったか」、「どこで当たったか」の情報を得る。その情報を元にオブジェクトに変化を起こさせている。

6 SSQ

6.1 映像酔いについて

映像メディアの進歩により、様々な種類の映像を視聴することができるようになった。こうした映像は視聴する人の健康に負の影響を及ぼすことがある。具体的には、光感受性発作、映像酔い、立体映像による眼精疲労である。映像酔いの症状は乗り物酔いとほとんど同じ症状が起こり、初期症状としては、めまい、倦怠感、ねむけ、顔面蒼白、冷や汗、胃部不快感などが在り、最終的には吐き気や嘔吐などの症状が出る。映像酔いの主な原因について、乗り物酔いは動揺病と呼ばれ、主に前庭器が受ける刺激によって生じる。前庭器とは、耳の奥にある平衡感覚をつかさどる器官で、三半規管と耳石器からなる。また、視覚も動揺病に影響し、酔いの軽減や酔いを強くする場合がある。映像酔いは乗り物酔いと違い、視覚刺激のみによって生じる動揺病である。原因はまだはっきりしておらず、視覚と前庭器の入力、あるいは三半規管と耳石器の入力が互いに矛盾すると、感覚の混乱が起こる説や、外界空間の誤った近くが平衡維持を脅かし、警報として不快症状が誘発されるという説がある。

6.2 SSQ

映像酔いによる生体影響の観測手法には、主観的評価などの心理的計測手法と、自律神経活動に関する観測などの生理学的計測手法がある。主観評価の計測で広く使われているものがSSQである。SSQとは多数のシミュレータ体験者から得た主観評価結果を因子分析することでシミュレータ酔いに有効と考えられる16の主観評価項目を抽出したもの。各項目を、4つの選択肢(なし、わずかに、中程度、激しく)から1つ回答する。それらの項目に1か0の加重をかけて加重和をとる。するとNausea(気持ち悪さ)、Oculomotor(目の疲れ)、Disorientation(ふらつき感)を示す評価値が求まる。この3つの評価値を集計すると、酔いの総合的な指標であるTotal sicknessscore(総合点)が得られる。SSQの特徴として一般の動揺病の評価と目の疲れに関する項目が含まれている点異なっている。

7 VRの類似技術

世の中にはVR技術以外にも仮想的な世界を作り出すという点においてVRと類似する技術があるので、それらを述べたいと思う。また、理解の手助けとなるように、各々の技術についてその具体的な例も挙げる。

7.1 VR技術

ここ以外でも述べられているが、VR技術とは「人間の感覚器官に働きかけ、現実ではないが実質的に現実のように感じられる環境を人工的に作り出す技術の総称」である。ここではあまり詳しく述べない。プレイステーションVRなどに応用されている。

7.2 AR 技術

AR は”Augment Reality” の略称で訳は「拡張現実」である。VR が完全に現実から切り離されているのに対し、AR では現実世界を AR 技術によって拡張しようというのが狙いである。拡張とはつまり、現実世界を舞台にしてそこに架空のものを創造することである。実用的な例としては、昨年大流行した Pokemon Go というゲームを思い浮かべてもらいたい。

7.3 DR 技術

DR は”Diminished Reality” の略称で、「消失現実」と訳される。AR 技術が現実世界に架空のものを付け加える技術であったのに対し、DR 技術によって実現されることは現実世界のものを消去することである。DR 技術の達成目標とは、物体を除去したことを体験者に分からせない、もしくはその処理過程を意識させないだけの高品質な視覚的除去を実時間で遂行することである。考えられる応用例としてあげられるのは、例えば、映画館で前の座席の人の頭を消すことなどが考えられる。

DR 技術の目標達成は難しい点が多いとされている。現在では画像・動画編集が手軽にできるスマホアプリやソフトウェアが沢山あるという状況を見ると、目に映るものを単に消すというだけならあまり技術的な知識がない人でも可能である。難しい点はその処理をリアルタイムに行わなければならないということである。

7.4 SR 技術

SR は”Substitute Reality” の略称で、「代替現実」と訳される。過去のある時点であらかじめ映像を撮影し、それをリアルタイムに映る現在の映像に混ぜて表示させるという技術である。例えば、B という人が映っている映像をあらかじめ撮影し、その数時間後、A という被験者が B のいる前で SR 技術を体験するとする。このとき、A は現実にいる B の姿と過去に撮影していた B の姿を同時に見せられることになるので、A にとってはあたかも B が二人いるように見えるのである。

7.5 MR 技術

MR は”Mixed Reality” の略称である。Mix という名の通り、現実世界と MR によって創造されたものが互いに影響を及ぼしあう。Magic leap という解者が多額の投資をしている。未だに謎の多い技術である。AR と似ている技術だが、その違いは AR が現実世界が主体であるのに対し、MR は仮想世界が主体であるということである。次の図に MR 技術の応用例を示した。この図では MR 技術は工場で用いられており、各生産ラインでの進捗情報が空中にポップアップする形で作り出されている。勿論、進捗が上がると対応する箇所の数値も変動する。このように、

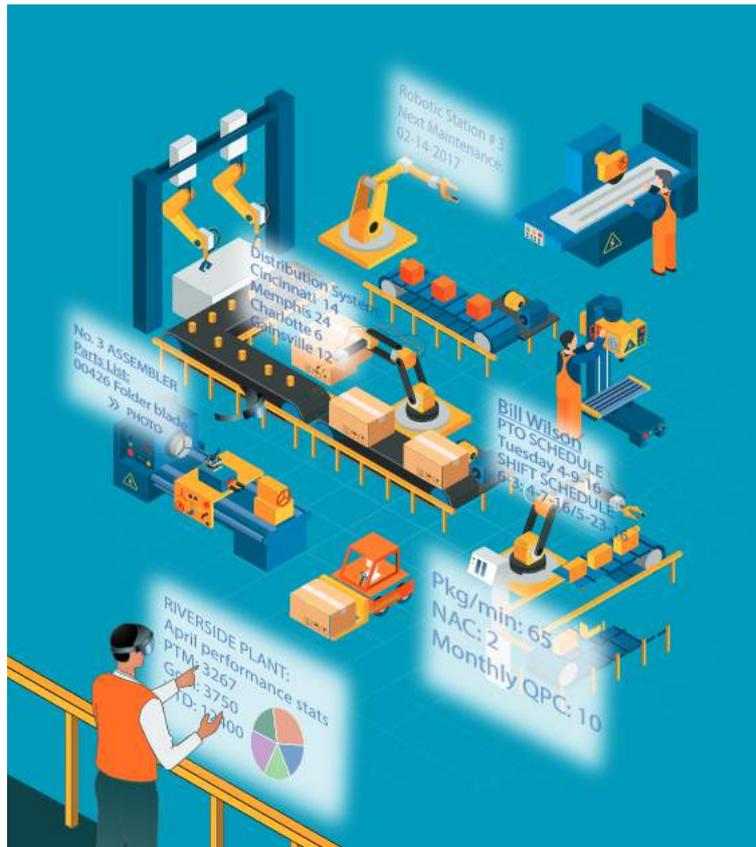


図 26: MR の工場における活用例

8 バーチャルリアリティ全般

本稿では、「バーチャルリアリティ」という語句自体について書籍「バーチャルリアリティ学」を参考に調べた内容を記述する。

8.1 バーチャルリアリティの3大要素

バーチャルリアリティには以下の3大要素がある。バーチャルリアリティは以下の3要素を含む。

- 3次元の空間性
- 実時間の相互作用性
- 自己投射性

8.1.1 3次元の空間性

これは普段自分たちが生活している空間と同等であることを示す。3次元の視覚空間、3次元の聴覚空間を持つことを指す。

8.1.2 実時間の相互作用性

現実における時間軸と、バーチャルリアリティ空間の時間軸が一致することをいう。バーチャルリアリティの世界が現実よりもゆっくり進んだり、あるいは高速に進んだりしないことを意味する。

8.1.3 自己投射性

現実での自分の動きがバーチャルリアリティ空間内の動きに反映される。例えば、現実の自分が腕を動かすと、バーチャルリアリティ空間内の自分自身も腕を動かす。

8.2 バーチャルリアリティの認知

バーチャルリアリティは、人工的に作り出したものであり、現実中存在するわけではない。それをあたかも本当に存在するもののように感じるができるのかという疑問がある。人は、感覚器官(五感)を通して対象を認識しているため、人が現実中存在する物を知覚する場合と同様の感覚を与えることで、バーチャルリアリティも実際に存在していると感じさせることができる。

8.3 バーチャルリアリティの用途

バーチャルリアリティは次の「3C」「3E」を満たすための道具であるにとらえることができる。

- Creation
- Control
- Communication
- Elucidation
- Education
- Entertainment

8.3.1 3C-Creation

Creation とは、創造である。「Tilt Brush」のように、バーチャルリアリティを用いて絵を描いたり、モノを作ったりすることを指す。



8.3.2 3C-Control

Control は、制御、操作にあたる。「da Vinci」のような、人の手の動きを機械に与えるロボットなどがこれにあたる。



8.3.3 3C-Communication

Communication はそのままコミュニケーションを指す。つまり現実に存在しない人などとのコミュニケーション、PlayStationVR で発売された、家庭教師になって VR 内の女の子を指導する「サマーレッスン」などがこれにあたる。

8.3.4 3E-Elucidation

Elucidation は解析を意味し、バーチャルリアリティを用いてシミュレーションや分析を行うことである。

8.3.5 3E-Education

Education は教育で、バーチャルリアリティを用いて教育を行う。

8.3.6 3E-Entertainment

Entertainment は娯楽で、バーチャルリアリティを用いた多くのゲームがこれにあたる。

9 まとめ、今後の展望

本活動を通して、VR コンテンツを作る為に必要な様々な技術について知ることができた。ただ単に VR と、VR 酔いの原因について理解を深めるだけでなく、VR の基盤となる CG や音響といった技術についても調べた結果、多くの知見が得られた。これらは今後 VR コンテンツを作成する場合以外にも役立つことがあるだろう。プロジェクトの途中段階では、最終的に作成した VR コンテンツと SSQ を用いて、どういっ

た要素が VR 酔いに起因しているかを検証しようとしていた。しかし開発, 調査の両面で間に合わず, それぞれの結果を発表する形となってしまう, 制作したコンテンツが実際にどの程度 VR 酔いを抑えられるかは評価できていない。また, 既存の VR コンテンツとの比較も行うつもりであったがそれも叶っていない。今後としては今回制作した経験, 得られた知識を活かし, 本会内で VR を用いた活動をより活発に行っていきたい。

参考文献

- [1] ベストプラクティス
<http://static.oculus.com/documentation/pdfs/ja-jp/intro-vr/latest/bp.pdf>
- [2] PenTile とは何なのか? GALAXY S3 のディスプレイを検証する。
<http://www.tjsg-kokoro.com/2012/05/25/pentile>
- [3] OpenCV と Visual C++による画像処理と認識 (20)
http://ishidate.my.coocan.jp/opencv_20/opencv_20.htm
- [4] レンダリングパイプライン
<http://www.wakayama-u.ac.jp/~tokoi/lecture/gg/ggnote01.pdf>
- [5] 変換 (2)
<http://www.wakayama-u.ac.jp/~tokoi/lecture/gg/ggnote04.pdf>
- [6] 陰影付け
<http://www.wakayama-u.ac.jp/~tokoi/lecture/gg/ggnote06.pdf>
- [7] テクスチャマッピング (1)
<http://www.wakayama-u.ac.jp/~tokoi/lecture/gg/ggnote07.pdf>
- [8] 説明しよう! シェーダーとは!!
http://d.hatena.ne.jp/o_healer/20110408/1302236008
- [9] 立体音響とインタラクション
<https://www.slideshare.net/quolc/ss-41439077>
- [10] uGUI を VR で使おう。 UI Builder #アセットアドカレ
<http://2vr.jp/2015/12/19/unity-assets-ui-builder/>
- [11] Unity 5.4 & 5.5 新機能キャッチアップ講座
<https://speakerdeck.com/unitydojo/unity-5-dot-4-and-5-dot-5-xin-ji-neng-kiyatutiatupujiang-zuo>
- [12] UNITY 5.5 がリリースされました!
<https://blogs.unity3d.com/jp/2016/11/29/unity-5-5-is-ready-for-you/>
- [13] Unity を Git でバージョン管理する
<http://qiita.com/hkomo746/items/432ed5b3a3395a804f5f>
- [14] 複数人で1つの Unity プロジェクトを管理するには
<https://thinkit.co.jp/story/2015/09/02/6362>

- [15] 映像が生体に与える影響の防止方法
<http://klibredb.lib.kanagawa-u.ac.jp/dspace/bitstream/10487/1490/1/kana-3-2-0003.pdf>
- [16] 自己運動感覚に影響を及ぼす高次視覚 情報の検討
<http://www.kochi-tech.ac.jp/library/ron/2013/2013info/1140294.pdf>
- [17] 知っておきたいキーワード 映像酔い
<https://www.ite.or.jp/contents/keywords/FILE-20111231153818.pdf>
- [18] 紅白の Perfume 演出に使われた「Dynamic Virtual Display(ダイナミック VR ディスプレイ)」がどうすごいのかわかるムービー
<http://gigazine.net/news/20170104-dynamic-virtual-display/>
- [19] バーチャルリアリティ
<https://ja.wikipedia.org/wiki/%E3%83%90%E3%83%BC%E3%83%81%E3%83%A3%E3%83%AB%E3%83%AA%E3%82%A2%E3%83%AA%E3%83%86%E3%82%A3>