

仮想環境と現実をシンクロさせた クロスサウンディングシステム開発

別府瞭¹ 川合康央¹

概要：本研究は、過去の街並み景観を再現するシミュレーションシステム上で、仮想環境と現実環境を可能な限り同期させることによって、知覚の差異によって発生する酔いを軽減させるためのシステムを開発したものである。天候と時間という人間が特に注意を払う部分を、仮想環境上で再現するシステム開発を行った。Windows から時間を、OpenWeatherMap API から天候データをリアルタイムに取得し、ゲームエンジンで用意しておいた季節と天候を呼び出すことによって、仮想環境に大きく変化をもたらすシステムを開発した。

1. はじめに

これまで本研究では、過去の歴史的な景観を再現するシミュレーションシステムとして、いくつかの開発を行ってきた[1,2]。開発したシステムは一定程度有用なものであったが、いくつかの課題が残った。本研究では、これらのうち、VR (Virtual Reality) 空間上での、現実空間との違和感について着目し、特に時間軸に関して、その軽減するシステムについて検討したものである。本研究では、過去の街並み景観を再現するシミュレーションシステム上で、仮想環境と現実をシンクロさせたシステムとして、開発を行った。現在、VR 研究で大きな壁となっているものとして、現実世界との差異が挙げられる。この問題は、仮想環境と現実世界で差が生じた際に、知覚した環境を処理する脳が混乱をきたしてしまい、VR 酔いという現象につながり、長時間仮想環境用でプレイをすることが困難になってしまうといったものである。乗り物酔いは、主に振れが原因で酔いが発生するが、VR ではシミュレータ（操縦訓練、その他特殊訓練）は、実体験との不一致で VR 酔いが発生し、汎用 VR では産業応用（ゲーム、設計、プレゼンテーションなど）や、医療応用（リハビリテーション、メンタルケアなど）、芸術表現に関してはやはり実空間上との予測の不一致などで酔いが発生している[3]。そこで本研究は、仮想環境と現実を可能な限り同期させることによって、ユーザの VR 空間への没入感を高めることを目的として開発を行ったものである。可能な限り現実空間と同期を行いたい、仮想空間のメリットを阻害してしまってはその意味がない。そこで、ここでは時間と天候に着目をするシステムとした。この二つの外的要素は人間が生活を行う上で重要な要素である。これらの要素を仮想空間内に取り入れることによって、HMD (Head Mounted Display) を被った時と外界との差を極限までに減らし、実体験との不一致を抑制できると考えた。

2. システム開発

2.1 時間経過

本研究で開発したシステムでは、没入感を意識して開発を進めることとした。その際に、時間をリアルタイムに同期させるというシステムを作成した。HMD を被った瞬間に時間が変わるというのは違和感がある。時間という概念は変わらないため、VR 空間の中でも現実の中でも時間は同じである必要性が、没入感を高めるために必須の要素である。そこで、WindowsPC 内の時計から時間を取得し、システム空間内の時刻に反映させることで、この矛盾点なくすることとした。また、システムの主要な光源である太陽の位置を緯度経度で計算し、気象庁による日射量観測から全天日射量を参考として、季節による太陽の明るさや、日没の速さなども実装した(図 1)。さらに雲量についても、OpenWeatherMap API[4]から取得し、日によって雲の量や、雲のスピードなどを調整した(図 2)。

2.2 季節変化及び天候

時間経過を実装する開発過程において、季節による日照時間も考慮する必要があることが分かった。そこで本システムでは、OpenWeatherMap API を使用して、10 分おきにリアルタイムに天候を反映させる手法を採用することにした(図 3)。本システムは、現在の天候をリアルタイムに判定し、あらかじめ用意してある 4 種類のマップを読みこむものである。次に、地面システムとして、雨天時に歩くとぬかるみを、雪が降っている際は雪に足跡が付くなど、天候により地面が受ける変化を再現した。

太陽システムは、Windows 内の時計から時間を受け取り、実際の太陽位置を計算し、日の出から日の入りまでの日射

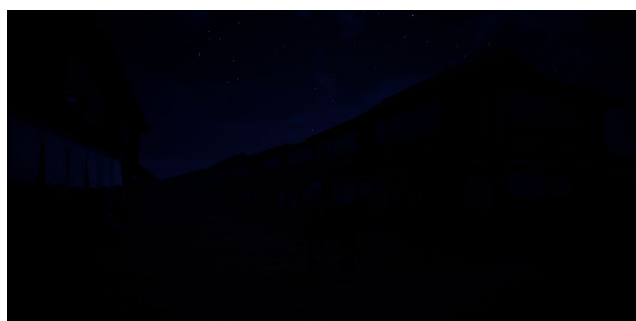
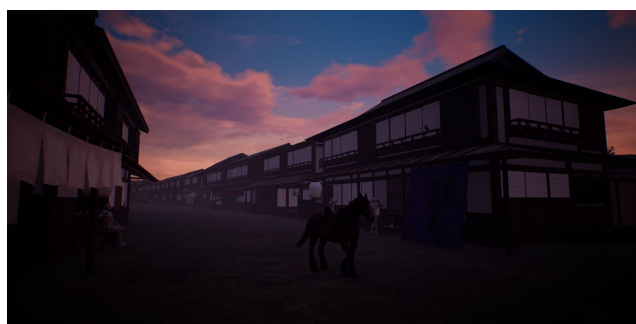


図 1 時間の同期, 上から 7 時, 12 時, 18 時, 24 時

量を計算するシステムである。また、季節によって日の入り、日の出が異なるため、これらを季節システムにより定めることができるものとした。季節変化は、気象庁が定めている「時に関する用語」[5]に則り、春は 3 月、4 月、5 月、夏は 6 月、7 月、8 月、秋は 9 月、10 月、11 月、冬は 12 月、1 月、2 月と定義して、リアルタイムに季節を反映させることとした。

また、天候に関しては、前述の OpenWeatherMap API を使用することによって、Unreal Engine 4[6]から 10 分おきに API をコールし(図 4)、JSON 形式で天候情報を受け取って、そのデータを基に天候を反映していくものとした。さらに、降水量や雲量も取得し、システム空間内に反映させることに



図 2 雲量, 上から 0 %, 50 %, 100 %

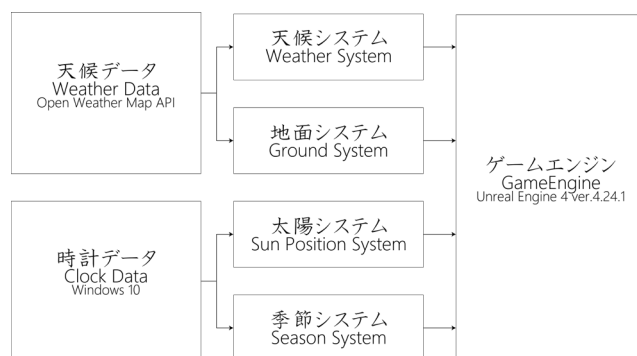


図 3 開発フロー

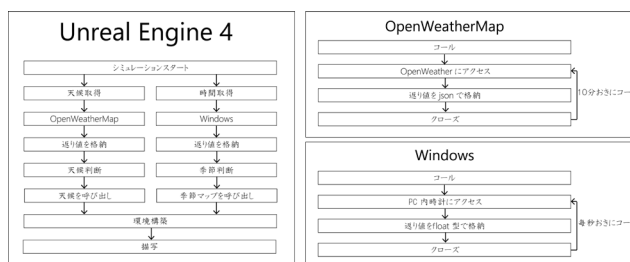


図 4 システム構成図

よって、現実との同期をさらに深めることとした。本来であればリアルタイムで天候同期を行いたいが、毎秒読み込むと処理が重くなったため、事前実験を行い、10分という単位を定めた。また、天候が切り替わった際は、即座に切り替わるのではなく、切り替えパーセンテージを設け0%から100%へと、徐々に切り替わるようにした。これは、即座に天候が切り替わった際、没入感への妨げになるため、ユーザがいつの間にか気が付くと変わっているといった程度の変化へと仕様の変更を行った。

晴天時と雨天時の違いとして、同時刻でも太陽が雲に覆われ暗いという点から、雲を増やした際、雲の上に行けば同じ日射量となるが、地上は少し暗いといった明るさの表現について再現を行った(図5)。また、雨天時の雨の描写においても、すべての場所に降雨エフェクトを配置すると処理が重くなってしまうため、プレイヤーが近づく範囲のみ描写を行うようシステムを組んだ。これにより、違和感なくかつ軽い処理で降雨の描写を行うことが可能となった。



図5 晴天時と雨天時の比較

2.3 インタフェース

本システムでは、インタフェースとして Oculus[7] Touchを用いることとした。本インタフェースを採用することによって、キーボードやゲームパッドを用いず操作可能となり、より没入感を高めるものとした。また、適宜に物理演算を与えたモデルを配置することにより、ユーザの入力でオブジェクトに触れたり、倒したりできるようにした。

視点の移動に関しても、連続した歩行ではなく、移動する瞬間にブラックアウトさせて瞬間移動する手法(図6)を、用いた。この移動手法によって、VR酔いが生じる際に感じる、

自分は移動していないがゲーム画面が移動してしまうことによる差異で酔ってしまうという現象を抑えた。操作方法は、ボタンを押して放すといった単純操作での移動が可能なインタフェースを採用し、初めて体験するユーザでも問題なく扱えることを確認した。

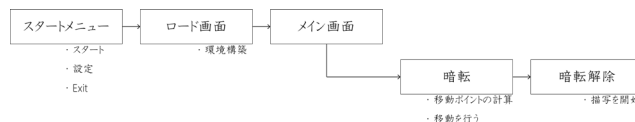


図6 画面遷移図

2.4 現在との比較

本システムを体験したテストユーザから、現在の同じ場所には何があるのかといった、現在との比較が欲しいというコメントがあった。そこで、現在に通じるオブジェクトを配置し、それに触れている間は現在の景観が見られるというシステムを取り入れた。現在の空間をあらためてモデリングを行うと、高いコストと重いシステムとなるため、今回は全天周画像を表示するといった Google Map に近い表現手法をとっている。現在の景観と比較することによって、地域の歴史に対する理解度が深くなることが考えられる。一方で、地元の人には場所が伝わりやすいが、初めて訪れる人には空間の位置同定が困難なため、対象地域の選定やミニマップなどのインタフェースを用意するなど、考慮を行う必要がある。

3. まとめ

本研究は、過去の景観を再現するシミュレーションシステム上で、仮想環境と現実を可能な限り同期させることによって、ユーザの VR 空間への没入感を高めることを目的として開発を行った。特に本稿では、仮想空間と現実を時間軸といった側面を中心に、同期させるといった試みを行ったものである。今開発では、仮想環境のメリットを軽減しない上で、調整することを目指した。今回、仮想環境と現実をシンクロさせるというテーマに沿うため、木などの環境配置物については、写真から生成されたフォトグラメトリのモデルを採用している。その際、単に配置するだけでは、処理に大きな負荷がかかってしまうため、LOD (Level of Detail) を設定し、モデルの精度を制御することにより、プレイヤーの視野範囲内のみ描写するなど、動作軽量化のための処理を施している。VR を扱う際に大きく FPS (Frames Per Second) 値が落ちてしまうと、身体動作との差異以外の理由で酔いが引き起こされてしまうため、処理によるフレームレートの低下をきたさないよう、システムの動作の軽量化を行った。

今後の課題として、天候は晴天、曇天、雨天、雪の4カテゴリに分類されるが、雲の量を測定し、より細かい、無雲、

晴れ, 雲が散在, 軽度の曇り, 部分的に曇り, 中程度の曇り, 大部分が曇り, ほぼ全天が曇り, 全天が曇り, 空の状態が不明という 10 カテゴリーに分けるといった, 国際気象通報式 [8]に変更を行いたい. このことで, 本システムを国内だけではなく, 海外のシステムにも取り入れることが可能となり, シミュレーションやゲームなど, その活用分野を大きく広げることが可能であると考え. また, ゲームエンジンを使用して開発を行ったため, 他のプロジェクトへの移行が簡単に行うことが可能である. この特徴を活かして, 様々な場面やシステム上で評価実験が行えることから, 今後, 各種条件を変更した種々のシーン上でのテストを経て, 汎用的なシステムとしての評価と考察を行っていきたいと考えている.

謝辞 本研究は, JSPS 科研費 No.19K12665 の助成を受けたものです.

参考文献

- [1] 川合康央, 土手光貴, 富樫尚繁, 鈴木隆太, 寺島雄大, 沼田和真, 藤井謙, 西野良祐. 江戸時代末期の旧東海道藤澤宿を対象とした街路景観シミュレーションシステムの開発. 情報処理学会インタラクシオン 2017 論文集, 2017, pp.726-731.
- [2] 川合康央, 大淵遥平, 土屋友梨夏, 二瓶翔子. 江戸時代末期の宿場町を対象とした歴史的街路景観シミュレーションシステムの開発と評価. 情報処理学会インタラクシオン 2018 論文集, 2018, pp. 690-694.
- [3] 中川 千鶴, 大須賀美恵子. VE 酔い研究および関連分野における研究の現状. 日本バーチャルリアリティ学会論文誌, 1998 年 3 巻 2 号, pp.31-39.
- [4] “Weather API – OpenWeatherMap”, <https://openweathermap.org/api>, (参照 2019-12-22)
- [5] “気象庁|予報用語 時に関する用語”, https://www.jma.go.jp/jma/kishou/known/yougo_hp/toki.html, (参照 2019-12-22).
- [6] “Unreal Engine 4”, <https://www.unrealengine.com/>, (参照 2019-12-22)
- [7] “Oculus”, https://www.oculus.com/?locale=ja_JP/, (参照 2019-12-22)
- [8] “気象庁. 国際気象通報式 (第 8 版) ”, https://www.jma.go.jp/jma/kishou/books/tsuhoshiki/kokusai/kokusai8_22.pdf, (参照 2019-12-22)