仮想会議の理解度向上を目的としたVR機能と表現方法の拡張の研究

本野克伎[1]　鶴田直之[1]

概要：本研究は，VR 機器を使用した三次元仮想空間でアバターを用いた会議システムを開発している．本稿では，HMD を活用した視線情報の取得機能に着目した，会議参加者の理解度向上を目的とした新たな機能実装を提案する．具体的には，取得した視線情報を基に聴講者がスライドのどこを注視しているかを発表者がリアルタイムで把握できる機能と，それらを累積したヒートマップを会議後に確認する機能とを実装した．発表者は，前者により視線情報を基に補足説明を加えたり，後者によりスライドの内容を改善したりすることで，発表の質を高めることができる．また，これらにより聴講者の発表内容への理解度向上が期待できる．実験では，発表者は視線情報を活用してスライドや発表内容の改善を行うことができた．

キーワード：仮想会議，プレゼンテーション，視線, HMD

Research on VR Functions and Representation Extensions to Improve Comprehension of Virtual Meetings

KATSUKI MOTONO[†1]　NAOYUKI TSURUTA[†1]

Abstract: This study develops a meeting system using avatars in a three-dimensional virtual space using VR equipment. In this paper, we propose a new implementation of a function that aims to improve the comprehension of meeting participants, focusing on the function of acquiring gaze information using a HMD. Specifically, we implemented a function that enables the presenter to grasp in real-time which part of the slide the audience is gazing at based on the acquired gaze information, and a function that allows the presenter to check the heat map accumulated from this information after the meeting. The presenter can add supplementary explanations based on the gaze information using the first function and improve the content of the slides using the second function, thereby enhancing the quality of the presentation. These can also be expected to improve the audience's understanding of the content of the presentation. In the experiment, presenters were able to use eye gaze information to improve the slides and presentation content.

Keywords: virtual meeting, presentation, gaze, HMD

1. 序論

1.1 研究の背景

近年のコロナ禍の影響により，自宅でテレワークやオンライン会議を行う機会が増加している．コロナ禍後の現在においても，対面よりも手軽に参加可能なオンライン会議や授業は珍しくなく，日常的に利用されている．

筆者らは，従来のオンライン会議や対面の会議に代わる選択肢として三次元仮想空間での会議システムが広く用いられることを念頭に置き，会議における参加者の心理的な側面の分析を目的としたアバター―ベースの仮想会議システムを開発している[1][2][3]．これまでに実装した主な機能は以下の通りである．

- 参加者のアバターの生成機能
- アバターの移動及び位置同期
- フォトリアル技術を用いた会議室のモデリング
- スライドや動画の共有機能
- ＶＲ機器での会議参加及びＶＲ用アバターの生成

このシステムでは，参加者は自分の分身となるアバターを操作し，音声を用いた会話やスライドを使ったプレゼンテーションを行うことができる．また，フォトリアルを活用してリアルな会議室を再現し，ＶＲ機器を使用してアバターを操作することで，会議への没入感を向上させることが可能となった．さらに，スライドや動画を操作する機能を搭載することで，対面のプレゼンテーションに近い体験ができるようになった．

1.2 研究の目的

本研究では，仮想空間で行う会議を「スライドを用いて発表者が聴講者にプレゼンテーションを行う会議」として定義する．本論文の目的は，仮想空間で会議を行うことで，対面の会議やオンライン会議と比較して，参加者の会議への理解度を向上させることである．

具体的には，聴講者の視線情報を可視化する機能を実装

1 福岡大学
†1 Fukuoka University

し，この情報を発表者が活用することで，スライドや説明内容を改善できるようにする．この発表改善機能により，発表者が聴講者に伝えたい内容をより効果的に伝達できるようになり，参加者全員の発表内容への理解度の向上を目指す．

2. 基本技術

2.1 Unity

本研究では会議システム及び三次元仮想空間（図 1）の開発に Unity を使用している．Unity は UnityTechnologies 社が提供するゲームエンジンや総合開発プラットフォームであり，主に C#言語でプログラムが記述される．

本システムは Unity でビルドしたファイルを起動することで，三次元仮想空間上に作成された会議室に入室することができるようになっている．

図 1　Unity 開発画面と仮想会議室

2.2 Photon・PRC 通信

Photon はオンラインゲームやアプリケーション開発に役立つゲーム開発用フレームワークであり，容易にマルチプレイヤー機能を実装できる．Photon のリアルタイム通信によるデータの同期が可能であり，本システムでは以下の目的で使用されている．

- 参加者の位置や音声データの同期
- オブジェクトの状態の同期

本システムでは，参加者間で同期を取るために RPC 通信を使用している RPC 通信（RemoteProcedureCall）はローカルで定義した関数を他のリモートマシンで実行させる仕組みである．この技術により，仮想空間内のオブジェクトの状態や動作を参加者全員で共有可能としている．

2.3 MetaSDK

MetaSDK は，MetaQuest 2 や MetaQuestPro などの VR デバイス向けアプリケーションを開発するための SDK である．本システムの VR 機能全般に使用させており，VR 用のカメラやアイトラッキング機能に使用されている．

2.4 アイトラッキング

アイトラッキングは，VR 機器に内蔵されたカメラを使用して，ユーザーの視線の動きや注視点をリアルタイムで追跡する技術である．本システムでは，この技術を活用してユーザーの視線情報を抽出し，発表者が聴講者の注視点を把握できるようにしている．

MetaSDK に含まれる OVRCameraRig を使用して視線情報を取得しており，この情報をもとに注視点を計算する[4]．また，VR 機器は MetaQuestPro を使用し，Meta Quest Link を用いて PC と接続し，VR 機器上で開発したシステムを起動して参加する仕組みである．

なお，アイトラッキング機能の利用には Meta Quest Pro 本体の開発モード設定が必要である．これにより，Meta Quest Link 経由でのベータ版アイトラッキング機能が利用可能となる．

3. 実装手法

3.1 システム概要

3.1.1 追加機能

本システムに大きく２つの機能を追加する．

１つ目は VR 機器を使用する使用者と使用しない参加者を同時に会議に参加可能とする機能である．先行研究で開発された VR 機器非対応のシステムでは，キーボード操作でアバターを操作するデスクトップ用システムが実装されていた．一方，VR 機器を追加したシステムでは，VR 機器がなければ会議に参加できない仕様であった．本研究では，VR 機器を使用しないデスクトップ参加者も会議に参加できるように改良を行った．

発表者がデスクトップで参加する場合，画面に映し出された発表補助情報を確認しやすい．一方，聴講者は VR で参加することで，視線情報をリアルタイムで取得可能となり，システムの活用効果が向上する．

２つ目はアイトラッキング機能である．VR 内臓カメラから視線情報を取得し，聴講者がリアルタイムでどこを見ているかを判別する．この視線情報は参加者間で通信・共有され，発表者に分かりやすい形で表示されるように実装した．

3.1.2 システム全体の流れ

システム利用の流れは以下の通りである（図 2 参照）．

① 発表者はデスクトップで，聴講者は VR で会議に参加する．

② 発表者が「開始」ボタンを押すと，アイトラッキング機能とタイマーの計測が開始される．

③ 発表者は聴講者の注視地点や，スライド注視時間の情報をリアルタイムで確認しながら発表を行う．

④ 発表者が「終了」ボタンを押すと，アイトラッキングとタイマーの計測が終了する．

⑤ 会議終了後，参加者の視線情報を基に作成されたヒートマップが出力される．

出力されたヒートマップを用いることで，発表者はさら

なる発表の改善を行うことができる．

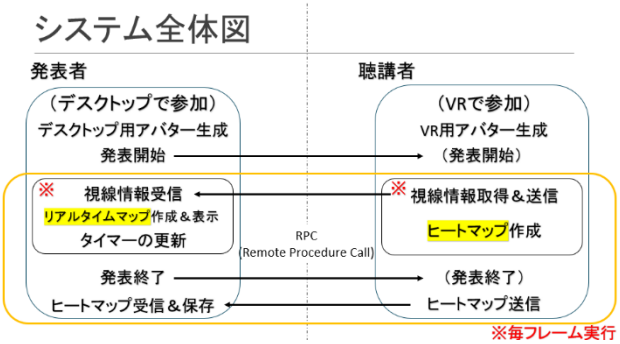

図 2　システム全体流れ

3.2 VR およびデスクトップ対応の参加方法

先行研究でのシステムは，名前を入力後にアバターが生成される仕組みを採用していた．本システムは VR とデスクトップの双方に対応するため，システム起動時に VR 機器の接続の有無に応じて，生成するアバターを分岐させる必要があった．デスクトップ用のアバターは先行研究で使用されたアバターを引き続き採用して名前を「unitychan_desktop」としている．一方 VR 用のアバターは先行研究時のアバターに以下の改良を行い，名前を「unitychan」とした．

- 名前入力を省略
- アバター生成時にレイヤーを"MyVRAvatar"に変更
- VR 用カメラが"MyVRAvatar"を映さない設定に変更
- 両手の位置に白い球体オブジェクトを配置

この変更で図 3 のように VR 参加者は自分のアバターを視認できないようになるが，他参加者から見ると図 4 のようにアバターが確認できる．これは VR 視点でのアバターの顔や体が視界に映り込むことによる視認性低下を防ぎ，アイトラッキングの精度を向上させている．

システムは if(UnityEngine.XR.XRSettings.isDeviceActive) を用いて，デスクトップ参加と VR 参加を判定している．VR 参加の場合は，「VR＋”会議参加人数」を名前として設定し，VR 用カメラや操作を可能にする”OVRPlayerController”をアタッチする．デスクトップ参加の場合は，名前を入力後にデスクトップ用カメラをアタッチしている．使用しないカメラは非表示にする仕様としている．

なおシーン上にデスクトップ用カメラと”OVRPlayerController”は最初から配置している．しかしアバターは Photon の同期の仕様上，指定されたフォルダ下にアバターを格納し，ファイル実行時に生成するため，起動時にシーン上に存在しない．そのため，アバターを生成した後に，対応するカメラをアタッチして使用しないカメラは非表示にしている．

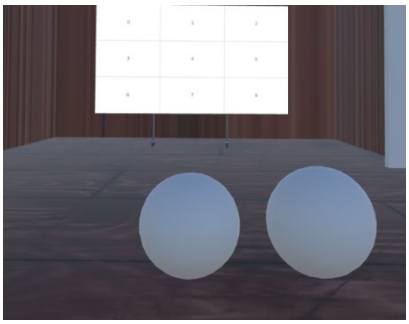

図 3　VR 参加者の両手の 1 人称視点

図 4　他参加者から見た VR 参加者

3.3 アイトラッキングの実装方式

視線情報の取得と利用を以下の２つに分けて実装した．

① EyeGazeController.cs： VR 機器内の視線情報を取得し，送信する役割を担う．

② PresentationManager： EyeGazeController.cs から送信された視線情報を処理し，タイマー管理や発表者用 UI への表示を行う．

3.3.1 EyeGazeController.cs

OVRPlayerController 内の OVRCameraRig を用いて視線情報を取得する．OVRCameraRig は以下のような階層構造を持っている．

```
OVRCameraRig
├── TrackingSpace
│    ├── LeftEyeAnchor
│    ├── CenterEyeAnchor
│    ├── RightEyeAnchor
│    ├── LeftHandAnchor
│    └── RightHandAnchor
```

TrackingSpace は空間トラッキングの基準点であり，各 EyeAnchor が視点のカメラの位置を表している．この視点の位置に” EyeGazeController.cs” を新規作成し，以下の処理を実行している．

① 視線ベクトルの作成

アイトラッキングにより視線の視点と方向が取得できる．これを用いて視線ベクトルを作成する．この処理は毎フレーム実行しているため，アバターが移動したり視線を移動したりすることで視線ベクトルも更新される．

② 視線ベクトルの衝突検知

視線ベクトルが当たり判定のあるオブジェクトに衝突したことを検知する．このとき衝突したオブジェクトが指定

したオブジェクト（＝スライド）を見ているかの判定も行っている．

③　注視地点の計算

スライドオブジェクトを見ている場合，衝突した座標から，スライドを九分割したときのどの領域をみているかを計算する．

④　RPC 通信による通知

「スライドを見ている」「スライドを見ていない」「注視点が導出できた」これらの処理が発生するたびにプレイヤー名と一緒に RPC 通信で参加者全員に送信している．

3.3.2 PresentationManager

シーン上に直接配置しており，以下の処理を実行する．

①　各種 UI の更新

会議全体の時間とプレイヤーごとのスライドを見ている時間を計測している．

②　リアルタイムマップの更新

聴講者がスライドのどこを注視していたかを，プレイヤーごとに色分けして発表者に表示している．EyeGazeController.cs で送信されたスライドと視線の衝突した座標を元に，表示するマップのサイズに変換しながらリアルタイムでマッピングを行っている．

このマップはスライドごとに更新されており，発表者は聴講者のスライドの注視した履歴が分かる（図 5）．

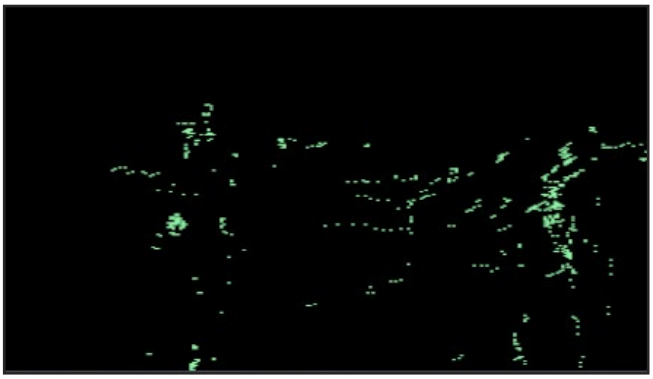

図 5　発表者用のリアルタイム表示マップ

③　ヒートマップの作成

会議終了後に発表者が分析するためのヒートマップを作成する（図 6）．②と同様に視線情報を元にマッピングを行っているが，②とは異なり自分の視線情報だけをマッピングしている．また，このヒートマップはシェーダーを利用しており，注視している場所ほど中心が赤くなるため，スライドのどこが重点的に見られたかを判別できる[5][6]．このヒートマップは会議終了とともに PC 上に保存される．自分のヒートマップを保存後，RPC 通信で byte 型として他の参加者に送信，他の参加者から受信したヒートマップの保存処理を行う．ファイル名は“スライド番号”_”プレイヤー名”.png と保存されるため誰がどのスライドをどのように見ていたかの確認ができる．

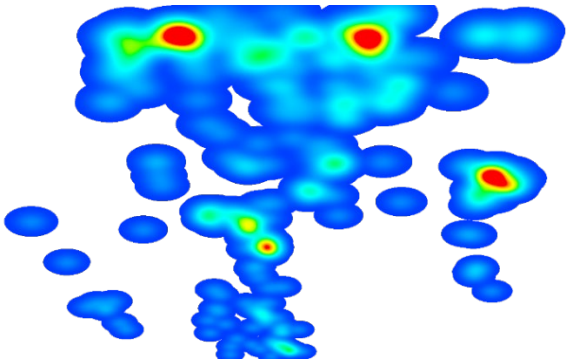

図 6　出力されたヒートマップ

4.　評価

4.1　評価方法・実験方法

本研究では，実装したシステムを評価するために，発表者と聴講者の２つの立場に分けて実験を行った．それぞれの評価ポイントは以下の通りである．

- 発表者：視線情報によって発表はやりやすくなったか
- 聴講者：被験者が見ている場所と計測した視線情報は一致しているか

被験者には，実験中または実験後にアンケートを通じてシステムに関する評価や意見を収集した．

4.1.1 発表者実験手順

被験者（５名）は同じ内容の発表を２回行う実験に参加した．実験手順は以下の通りである．

1. 被験者は，筆者が事前に準備したスライドを使用して発表を行う．このとき，スライドのみを見ながら発表を行う．
2. 発表終了後，被験者は聴講者のヒートマップを閲覧し，それをもとに発表の改善計画を考案する．
3. 改善計画を基に，２回目の発表を行う．この際，リアルタイムで聴講者の視線情報を閲覧しながら発表を行う．

4.1.2 聴講者実験手順

被験者（９名）は１度発表を聴講し，視線情報の正確さを確認する以下の手順で実験を行った．

1. 被験者にはアイトラッキングの詳細を伝えずに，発表を聴講させる．
2. 聴講後，アイトラッキングについて説明し，視線情報の可視化について理解を深めてもらう．
3. リアルタイムでスライドの見ている場所が光る設定にし，実際に見ている場所と光の位置にずれがないか比較する．

4.2 実験結果

4.2.1 発表者実験の結果

実験後アンケートの主な数的データは以下に示す.

- 「リアルタイムに閲覧できる情報は発表の改善に活かせたか」
  - 大いに活かせた: 20%
  - やや活かせた: 80%
- 「２回目の発表は１回目の発表よりうまくできたか」
  - 満足のいく発表だった: 20%
  - やや満足のいく発表だった: 80%
- 「視線情報は発表中に閲覧できたほうがいいか」
  - 閲覧できた方がいい: 60%
  - どちらかといえば閲覧できた方がいい: 40%
- 「視線情報は発表後に閲覧できた方がいいか」
  - 閲覧できた方がいい: 100%
- 「発表する立場として視線情報をプレゼンターが得られる発表システムはどう思うか」
  - 導入されていくべき: 100%

１回目の発表後，ヒートマップを基に被験者が考案した主な分析と改善案は以下の通りであった.

- 分析
  - 文章に視線が集中している傾向がみられる
  - 画像がほとんどみられていない
- 改善案
  - 写真に説明を加えたり風景写真を説明しやすい建物の写真に差し替えたりする
  - スライドの文字数を減らし，発表者の口頭説明を増やす.

本システムへの具体的な意見や感想は次の通りである.

- 聴講者が説明している場所を見ているか確認できた.
- 見られていない箇所をリアルタイムで補足できた
- 次の発表にむけた具体的な改善につながる
- 挿入した画像や文字の効果を視線情報から具体的に評価できる

一方，他のデバイスで作成されたヒートマップが正常に保存できないという課題が生じた．初期数枚は保存できたが, 以降のデータは欠損が生じた. 例えば図 7 は参加者 VR５の PC に保存された画像であり，VR５自身のヒートマップはスライド番号１から７まで正常に保存できている．対して, 一緒に参加していた VR3 のヒートマップはスライド番号１から３までしか保存されていない.

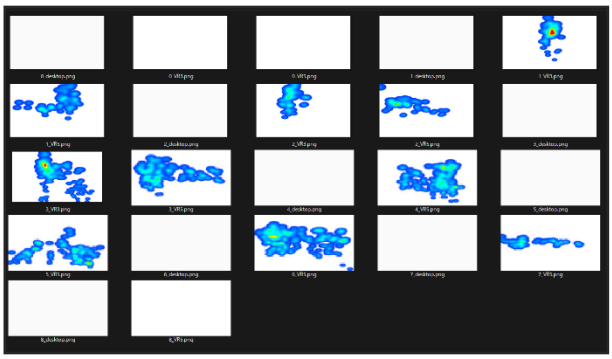

図 7　欠損した保存画像

4.2.2 発表者実験の考察と評価

視線情報を基にした発表改善は，全被験者で一定の成果を挙げた．特に１回目の発表後に文字に視線が集まりやすい点を分析し，２回目の発表では画像に説明を追加することで視線の誘導に成功した被験者が多かった．一方で，リアルタイムの視線情報閲覧について，発表者全員が肯定的であったが，そのうち 4 割は「どちらかといえば閲覧できたほうがよい」と回答した．発表中に視線情報を処理することに難しさを感じた被験者もおり，リアルタイム閲覧が常に発表の有利になるとは限らないことが示唆された.

前述のヒートマップの課題は，自分のデバイスで作成したヒートマップは正常に保存できていた点から，RPC 通信による保存処理の遅延や競合が考えられる．実験ではシステム外で画像の共有を行うことで発表の改善自体には影響はなかったが，システム内で完結するのが望ましい．そのため今後の課題として，効率的な通信処理の実現やデータ共有の安定性の向上が挙げられる.

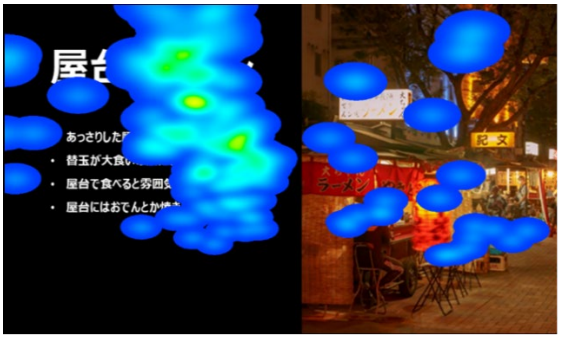

図 8　発表１回目のヒートマップ

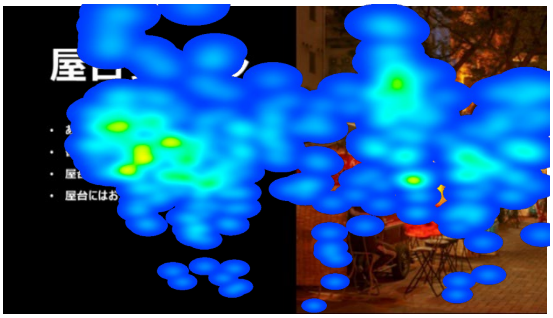

図 9　発表２回目のヒートマップ

4.2.3 聴講者実験の結果

目の追従については，ずれていたと回答した人は右下にずれていると回答していた．また，眼鏡の有無とずれの解答に相関は見られなかった．

以下にアンケートの数値評価の結果を示す．

- 「実際に見ている場所と表示された光の位置は一致していたか」
  - 一致していた: 55%
  - ややずれていた: 45%
- 「表示された光は目の動きについてきていたか」
  - 完璧についてきていた: 55%
  - やや遅れてきていた: 45%
- 「ヒートマップは目の動きを記録できていたか」
  - 記録できていた: 66%
  - おおよそ記録できていた: 34%
- 「ヒートマップは自分の想像と一致していたか」
  - 一致していた: 45%
  - おおよそ一致していた: 55%
- 「発表を聞く立場として視線情報をプレゼンターが得られる発表システムはどう思うか」
  - 導入されていくべき: 100%

その他，自由記述の回答を示す．

- 本システムへの具体的な意見
  - プレゼン自体に興味があるか伝わるから導入してほしい
  - 説明されていない，確認していないというトラブルを未然に防げる
  - 現実でされるプレゼンと特に差を感じなかった
  - 集中度や興味をプレゼン資料に反映させてほしい

4.2.4 聴講者実験の考察と評価

聴講者の視線情報は概ね正確に取得されており，発表者が発表の改善に役立てることができた．しかし視線情報の位置が若干右下にずれているという指摘が複数あった．考えられる原因として以下が挙げられる．

1. VR ヘッドセットの装着位置の個人差

ヘッドセットを浅く装着していたり，装着位置がずれていたりすることで，視線のキャリブレーションが正確に行われないことがあり，視線位置が正しい方向からずれる可能性がある．

2. 瞳孔間距離の個人差

実験では VR ヘッドセット内の瞳孔間距離の設定を統一して行っており，その設定が被験者に合っていなかった場合は視線の検出に誤差が生じる可能性がある．

3. 視線トラッキングハードウェアの精度の限界

VR ヘッドセットの視線トラッキングセンサー自体の精度に限界があり，細かい検出が難しい可能性がある．

4. 被験者の身長差

被験者の身長や視線の高さが異なるため，仮想空間内の視線位置が実際よりもずれていた可能性がある．

5. 計算アルゴリズムの誤差

本システムはアバターの目の位置を始点とした視線ベクトルで衝突の検知を行って，スライドとアバターの距離が遠いほど検出に誤差が出る可能性がある．

これらの原因に対処するため，会議参加前に参加者の身長でアバターの身長を調整する機能や視線のキャリブレーション機能を実装する必要を感じた．

また「仮想空間で発表を聞くことは現実空間で発表を聞くことと大差なかった」という意見が出た半面，VR をかぶることで少し酔ったという意見もでた．VR をかぶりながらの会議は酔い対策を取り入れたり，長時間の会議時間を想定したテストを行ったりする必要もあると考えられる．

5. 結論

本論文では，仮想会議の理解度向上を目的とした，VR を利用した視線情報の取得とスライドを用いた会議への応用システムを提案・実装した．

本システムでは，先行研究で構築された VR 機能に加え，内蔵カメラを用いた視線情報取得機能を実装し，リアルタイムで聴講者がスライドのどこをみているか把握できる仕組みを導入した．これにより，発表者が得た視線情報を基に説明やスライド内容を改善することで，参加者全体の会議における理解度向上を促進できることを確認した．また，このシステムは発表の練習として使用しても活用でき，視線データを基にスライドや発表内容を改善することで，現実での発表をより効果的なものにする可能性が示唆された．

一方で実験により以下の課題が明らかとなった．

1. 視線情報のわずかなずれ
2. ヒートマップ共有の課題

参考文献

[1] 松永祐奈，阿比留幹大，日下部祥基，本野克伎，鶴田直之，心理学研究への利用を目的とした 3 次元仮想プレゼン環境の試作，火の国情報シンポジウム 2023
[2] 内田 龍成，”３次元仮想会議システムにおける没入感向上に関する研究”，福岡大学工学部電子情報工学科卒業論文(2024 年)
[3] 山田 朋史，”心理分析用３次元仮想会議室におけるプレゼンテーション機能の高度化に関する研究”，福岡大学大学院工学研究科電子情報工学専攻修士論文(2024 年)
[4] Meta XR カメラ設定 https://developers.meta.com/horizon/documentation/unity/unity-ovrcamerarig/?locale=ja_JP
[5] Unity の Plane 上に注視点ヒートマップを作る https://qiita.com/sakamo1290/items/9edd0de46480c74ea078
[6]【HTC 開発者交流会】VIVE Pro Eye で視線ヒートマップを描く https://umehashi.hatenablog.com/entry/2022/02/08/090000