

学士特定課題研究論文

全天周ビデオ通信端末を用いた非対称型 コミュニケーションに関する研究

栗岡 保

17B05518

東京工業大学
情報理工学院
情報工学系

指導教員 小池 英樹

2021 年 1 月

概要

学士特定課題研究論文は、シングルカラムでページ数に制限はない。

目次

概要	ii
第 1 章 序論	1
1.1 本章の概要	1
1.2 本研究の背景	1
1.3 本研究の目的	3
1.4 本論文の構成	3
第 2 章 関連研究	4
2.1 360 度カメラに関する研究	4
2.2 全天周球状ディスプレイに関する研究	4
2.3 ビデオ会議に関する研究	4
2.4 まとめ	5
第 3 章 球体ディスプレイを用いた視線共有システムの提案	6
3.1 従来型ビデオ会議アプリケーションの問題点	6
3.2 提案する視線共有システム	6
3.3 技術的課題	7
第 4 章 OmniEyeBall 対 PC 間通信の実装	8
4.1 ハードウェア実装	8
4.2 ソフトウェア実装	8
第 5 章 実験	12
5.1 実験の目的	12
5.2 実験用アプリケーション	12
5.3 実験手順	12
5.3.1 実験設定	12
5.3.2 実験項目	13
5.3.3 被験者の動き	14
5.4 実験結果	15
5.4.1 定量的評価	15
5.4.2 定性的評価	15
5.5 実験結果から得られる知見と考察	16

第 6 章	考察	18
6.1	本システムの問題点	18
6.2	本システムの応用	18
6.2.1	対話相手表示形式	18
6.2.2	横顔生成形式	19
6.2.3	おかしら会議形式	19
6.3	今後の展望	20
第 7 章	結論	21
	謝辞	22
	参考文献	23
	参考文献	24

図目次

1.1	zoom	1
1.2	全天球パノラマ画像	2
1.3	theta V	2
4.1	OBSStudio で画面をキャプチャする様子	9
4.2	zoom での OBS-virtual-cam の映像ストリーミング	9
4.3	正距円筒図から正距方位図への変換	11
5.1	画像が移動する模式図	13
5.2	プレゼンターに用意したスライドの一部	14
6.1	OmniEyeBall 上で, PC 使用者の表示位置が移動する模式図	19

表目次

4.1	OBS-virtual-cam の仕様	9
5.1	質問項目の一覧	15
5.2	平面ディスプレイ使用時	16
5.3	OEB 使用時	16
5.4	5 段階評価項目の結果	16

第 1 章

序論

1.1 本章の概要

本章では、本研究の背景、目的、本論文の構成について述べる。

1.2 本研究の背景

近年、様々なビデオ会議アプリケーション (図 1.1) が登場している。その例としては、zoom [1] や Google Meet [2] が挙げられる。新型コロナウイルス感染症の流行により、ビデオ会議の需要がさらに高まりつつある。

しかし、ビデオ会議における様々な問題点も指摘されている。例えば Roel [3] は、カメラの視覚外の情報や、人物の情報の不足のために、対面時のようなインタラクションを得られないことを指摘している。



図 1.1 zoom

[1]



図 1.2 全天球パノラマ画像



図 1.3 theta V

[4]

一方で、昨今は誰でも気軽に全天球映像 (図 1.2) を撮影することができるようになっている。その例として、全天球カメラ (360 度カメラ, 全天周カメラ, 全方位カメラなどともいう (図 1.3)) を使用して、全天球映像をヘッドマウントディスプレイを用いて観覧したり、パノラマ映像として動画や静止画を保存することが出来るようになっている。

全天周カメラの使用により、カメラの視野の問題は解決される。実際に、Anthony ら [5] は、全天球の視野の広さによって、リモートユーザーがローカルユーザーの環境をより早く理解できると結論付けている。だが、Johnson ら [6] によって、パノラマ視野によって映像の複雑さが増し、より大きな認知負荷を必要としたことも示されている。

認知負荷を削減するため、パノラマ映像を 3 次元に表示する、球体プロジェクターを用いた方法が

挙げられる。球体プロジェクターの利用としては、Li らの OmniEyeBall

1.3 本研究の目的

1.4 本論文の構成

第 2 章

関連研究

2.1 360 度カメラに関する研究

- RICOH THETA V(<https://theta360.com/ja/about/theta/v.html>)
- insta 360(<https://www.insta360.com/jp/product/insta360-onex2>)
- Meeting OWL(<http://meetingowl.jp/?i=nav>)
- Collaboration in 360 属 Videochat: Challenges and Opportunities(<https://dl.acm.org/doi/10.1145/306460>)

2.2 全天周球状ディスプレイに関する研究

- Glomal350(<https://www.aisan.co.jp/products/glomal350.html>)
- WORLDEYE(<https://www.gakkensf.co.jp/worldeye/sp/>)
- OmniEyeball: An Interactive I/O Device For 360-Degree Video Communication(<https://dl.acm.org/doi/10.1145/3279778.3279926>)
- Qoom: An Interactive Omnidirectional Ball Display(<https://dl.acm.org/doi/10.1145/3126594.3126607>)
- Comparing flat and spherical displays in a trust scenario in avatar-mediated interaction(<https://dl.acm.org/doi/10.1145/2556288.2557276>)

2.3 ビデオ会議に関する研究

- Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment(<https://dl.acm.org/doi/pdf/10.1145/2818048.2819965>)
- Improving visibility of remote gestures in distributed tabletop collaboration(<https://dl.acm.org/doi/10.1145/3379350.3416197>)
- Putting things in focus: establishing co-orientation through video in context(<https://dl.acm.org/doi/10.1145/3139131.3141775>)
- A gaze-preserving group video conference system using screen-embedded cameras(<https://dl.acm.org/doi/10.1145/3139131.3141775>)
- Towards Enabling Eye Contact and Perspective Control in Video Conference (<https://dl.acm.org/doi/10.1145/3379350.3416197>)

2.4 まとめ

第 3 章

球体ディスプレイを用いた視線共有システムの提案

3.1 従来型ビデオ会議アプリケーションの問題点

ビデオ会議では、視野の狭さにより、参加者同士の状況が分かりづらいという欠点がある。そのため、コミュニケーションが円滑に行われない場合がある。全天球ビデオカメラを使用することで視野の狭さの問題が解決するが、一方で映像が複雑になり、その影響でやはり快適なコミュニケーションが行われない可能性がある。

一方で、先ほど述べた OmniEyeBall に代表される、全天周ディスプレイを用いたビデオ会議が考えられる。全天周ディスプレイでは、全天球映像が球を用いた立体的かつ自然な形で表現され、全天球映像を視認しやすくなる。

しかし、上記の方法でなお、視線情報の問題が存在する。従来のビデオ会議では、参加者がどの参加者を見ているかなどの視線情報が伝わらない。以下では、PC の使用者 1 人と OmniEyeBall の使用者複数人の非対称な通話状況において、PC 使用者の視線情報を伝える方法を提案する。以下で提案したシステムを用いて、従来のビデオ通話アプリケーションを用いた場合よりも PC の使用者の視線情報が OmniEyeBall の使用者に正確に伝わることを最終的な目標である。

3.2 提案する視線共有システム

PC 側で提案するシステムを以下に述べる。平面ディスプレイ上に OmniEyeBall に取り付けた全天球ビデオカメラのパノラマ映像を表示する。映像上で、参加者の顔をクリックすることで、その参加者が正面に来るようにパノラマ映像を平行移動させる。そのようにして、常に正面を見ていれば見たい人を見ているという状態を作り出す。

(ここに、画像をスライドさせる図を乗せる)

以降は、OmniEyeBall 側で提案するシステムを述べる。PC 側で使用する全天周ビデオカメラの映像を球体ディスプレイ上に表示する。PC 側のクリック操作に合わせクリックされた人物の方向に顔が向くように映像を回転させる。こうすることで、PC 側で見ている人間とみられた人間が向き合う状態を作る。

この時、例えば 3 人が 120 度の間隔で OmniEyeBall の周りに座っている状況を考えると、顔を向

けられていない2人は、ほとんどPC側の参加者の顔を視認できない。このような場合、この2人には画面上部に小さなウィンドウを表示し、そこにPC側参加者の顔部分を表示する。顔部分を表示すると、その顔は大抵正面方向を向いており、自身が見られていると錯覚する恐れがある。それを防ぐため、PC参加者の顔ウィンドウの隣には、見られている参加者の顔ウィンドウを表示させる。

(ここで図の挿入)

また、このままではPC側の使用者が、特定の個人を見ていない場合(例えば、全員に対して語りかける場合)を区別できない。このため、特定の個人を見ていない場合は、全員に対して画面上部にPC参加者のみの顔ウィンドウを表示させる。クリックした人物の方向を向く仕様では、常に前回のクリックの結果が残り、いずれかの人物の顔方向にPC参加者の顔が向くことになる。この方向に対して、あえて顔ウィンドウを顔映像に被せることによって、大きく表示される方の顔映像の視認性を下げ、自身が見られているという錯覚を防ぐことが狙いである。

(ここで図の挿入)

3.3 技術的課題

ハードウェア部分の実装に関しては、全天球ビデオカメラ2つ、PC2台、及びOmniEyeBall1台を用意すればよい。

一方、ソフトウェア部分の実装に関しては、いくつか準備の必要がある要素が存在する。

まず、ビデオストリーミングの方法について考える必要がある。UDP等を用いて映像情報の送受信を行う必要がある。しかし、フレーム落ちの問題など、現在使用されているビデオ会議アプリケーションのような堅牢で高速な通信を実現するためには、高度な技術と時間を要する。そこで、映像送受信については、すでに利用されているビデオ会議アプリケーションを用いることにした。

しかし、ただ映像を送受信しては、PC側の操作に合わせて映像を回転させるといった処理が出来ない。PC側では全天球ビデオカメラの映像を加工した後に送信を行う必要がある。あるいは、PC側で表示するUIにおいて、OmniEyeBall側の全天周パノラマ映像をそのまま表示させるのではなく、やはりクリックした人物の顔を正面に移動させるなどの必要がある。ここで、加工した映像をキャプチャしてビデオ会議アプリケーションに認識させる方法と、一方で、送信されてきた映像をキャプチャーして、以下で説明する画像加工ライブラリで扱えるようにする方法を考えなければならない。

この両方を解決するために、今回は仮想ビデオカメラを用いる。仮想ビデオカメラとは、OBS-virtual-cam [7] や UnityCapture [8] 等のような、特定のアプリケーションの映像をリアルタイムでキャプチャし、その映像をあたかも接続したビデオカメラの映像のように扱えるものである。これを用いることで、ビデオ会議アプリケーションや画像加工ライブラリが、加工済みの映像や送信されてきた映像をビデオカメラの映像として認識し、ストリーミングや加工が出来るようになる。

(以下に、映像送受信 + 加工のフローの図を乗せる)

次に、映像の加工方法について述べる。映像の加工はpythonなどで利用可能な画像加工ライブラリを用いた。同時にPC画面に表示するUIもpythonのライブラリで作成した。

第 4 章

OmniEyeBall 対 PC 間通信の実装

4.1 ハードウェア実装

全天周ビデオカメラ

全天周ビデオカメラとして, PC 側では RICOH THETA S, OmniEyeBall 側では RICOH THETA V を用いた.

球体ディスプレイ

全天球動画の出力装置としては渋谷光学の球体プロジェクターである Glomal350 を使用する.

4.2 ソフトウェア実装

映像ストリーミング

全天球映像の送受信には Zoom Video Communications の Zoom [1] を使用した. (メモ: 解像度と fps を後で調べる)

仮想カメラ

映像のキャプチャ手段として OBS Studio [9] を用いている. OBS Studio は起動中のアプリケーションや画面全体, 接続されたカメラの映像をキャプチャー・合成して作成した映像を Youtube 等の動画配信サイトへストリーミングできるアプリケーションである. 以下の図のタイマーのキャプチャのように, キャプチャしてきたアプリの映像にエフェクトを加えることも可能である. (映像では青色のクロマキーフィルタを加え, 背景を透明にしている)

仮想カメラとしては, OBS-virtual-cam [7] を用いた. この仮想カメラの仕様は以下のようになっている.

OBS-virtual-camera は OBS Studio の機能であり, これを用いれば OBS Studio でキャプチャ中の映像を, 仮想カメラの映像として扱うことが出来るようになる. 仮想カメラは, 他アプリケーションから PC に物理的に付属したカメラと同様に認識され, ストリーミングや映像の加工などが容易になる.

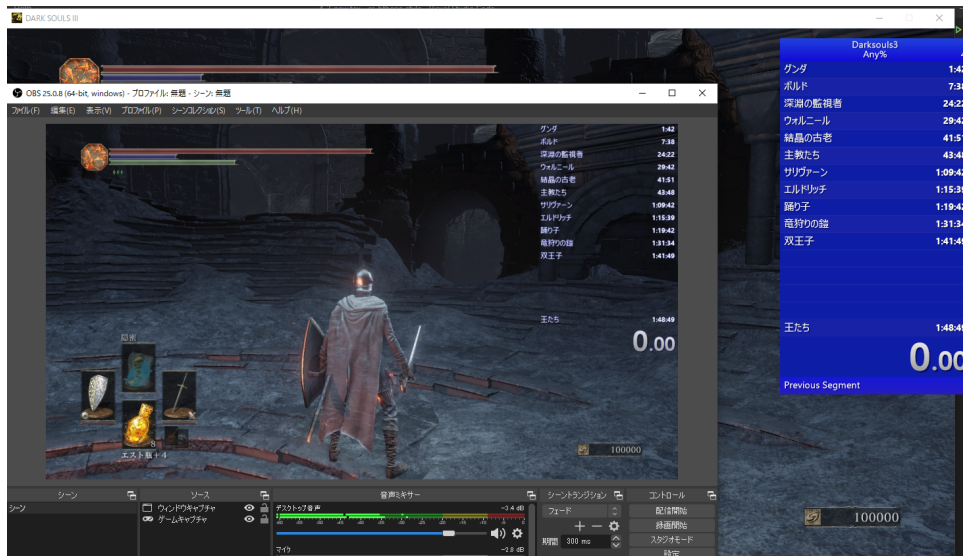


図 4.1 OBSStudio で画面をキャプチャする様子

最大解像度 1280x720

最大 fps 60

表 4.1 OBS-virtual-cam の仕様

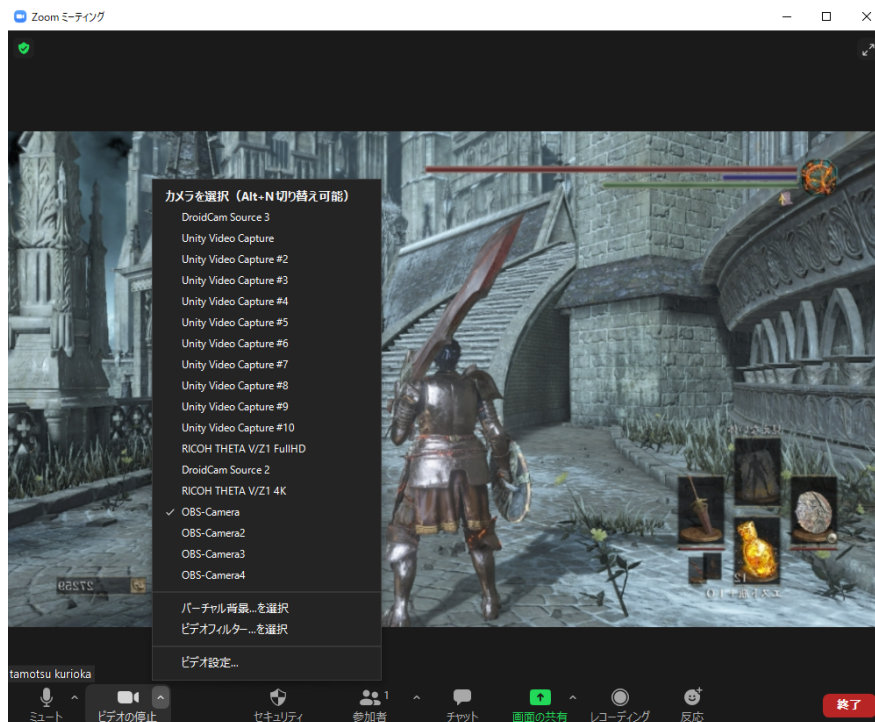


図 4.2 zoom での OBS-virtual-cam の映像ストリーミング

映像の加工

映像の加工には、python のライブラリの 1 つである OpenCV を用いた。先述した通り、全天球ビデオカメラのパノラマ映像は正距円筒図法を用いて出力されている。glom350 を用いる際にはさらに正距方位図法に変換する必要がある。

縦 H 、横 W の大きさで表された正距円筒図の点 $P(x, y)$ が半径 R の正距方位図で点 $P'(X, Y)$ に写されるとすると、以下の関係式が成り立つ。

$$\begin{aligned}\theta &= \frac{2\pi x}{W} \\ r &= \frac{Ry}{H} \\ X &= r \cos \theta \\ Y &= r \sin \theta\end{aligned}$$

この関係式を用いることで、全天球ビデオカメラで撮影したパノラマ画像を、glom350 で投影するための正距方位図法を用いた画像へと変換することが出来る。

GUI の構築

アプリケーションを操作しやすいように GUI を構築した。GUI の構築には、python のライブラリである Tkinter を用いた。

Tkinter では、他の GUI アプリケーション制作ライブラリと同じように、ボタンや画像などを表示させることが出来るほか、画像にカーソルを重ねたときに発生するイベント等を登録することができる。今回は、顔をクリックすると画像が平行移動する処理や、実験の解析のために操作のログを残すために使用した。

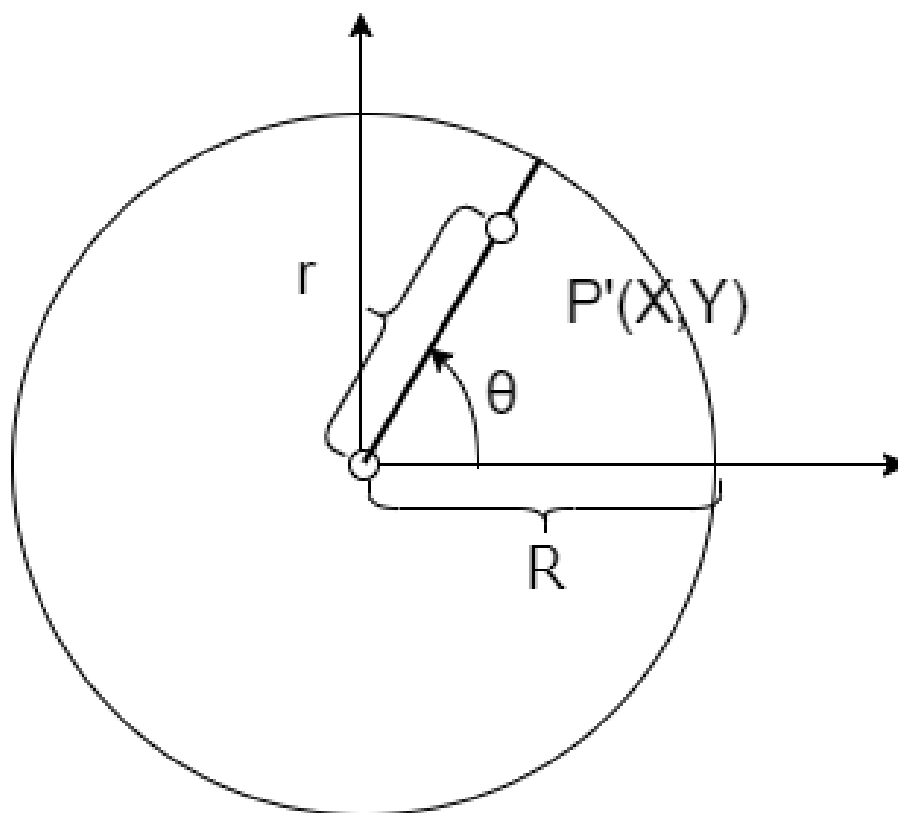
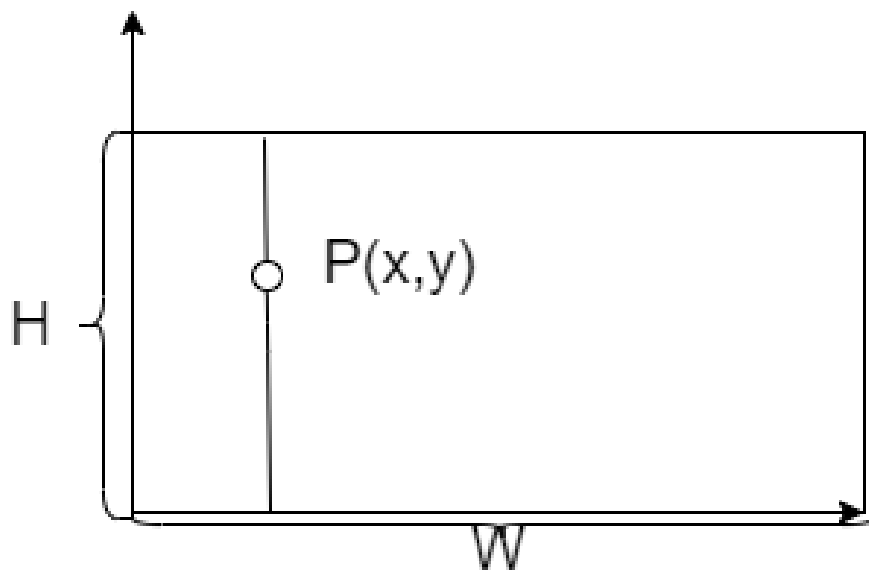


図 4.3 正距円筒図から正距方位図への変換

第 5 章

実験

5.1 実験の目的

OmniEyeBall 及び、4 章で紹介した視線共有手法を用いて、1 人が多人数に対しプレゼンテーションを行う際に被検者がどのように感じたかを定性的に調べるのが本実験の目的である。

一方で、意図した視線が正確に伝わっていたかを、視線情報の履歴から算出し、定量的な評価も行う。

定量的データとアンケート結果から、本システムの優れた点と改善すべき点を明らかにし、今後の研究へとつなげる。

5.2 実験用アプリケーション

- ・(4 章と重複? 要修正)
- ・画面のレイアウトについて詳しく記述する

5.3 実験手順

5.3.1 実験設定

プレゼンター 1 人、傍聴者 3 人によるプレゼンテーション実験を行った。

プレゼンテーション、及び質疑応答の時間合わせて 10 分程度の時間を設けた。

従来の平面ディスプレイでの表示を用いた実験、及び 5-2 節で説明したアプリケーションを用いた実験の 2 通りを行った。本来はカウンターバランスをとるため、逆順で 2 回行う予定だったが、コロナ感染対策のため参加人数を最小限に抑える必要があり、今回は 1 回での実験となった。

プレゼンターの負担の削減のため、プレゼンテーションのための資料は事前に作成した。2 通りの実験のため、2 種類の資料が存在する。プレゼンテーションの内容は、SNS 用の宣材写真として東京工業大学の敷地内で何を写真に撮るのか、という題で本館、または図書館を紹介するという物であった。

また、各傍聴者には、それぞれ最低 1 回の質問機会を設けるため、各プレゼンテーションに対して 3 つの質問項目を事前に用意した。

実験環境は以下の通りであった。

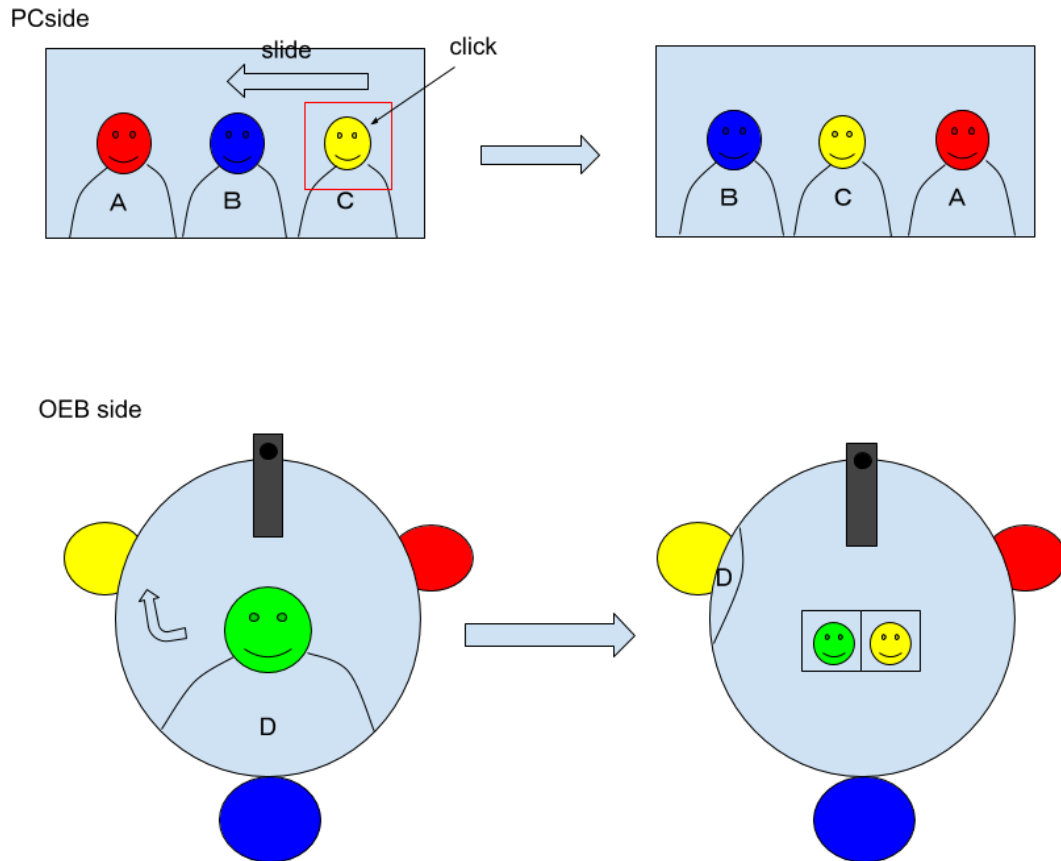


図 5.1 画像が移動する模式図

- プレゼンターの全天球カメラとして THETA S, 傍聴者側には THETA V を用いた
- プレゼンターと傍聴者は全天球カメラから約 40cm ほど離れていた
- (プレゼンターと傍聴者側で用いた PC の詳細を後で書く)
- 傍聴者は平面ディスプレイを使用する際は, PC の前に設置した全天周カメラの前で横に並んで座っていた
- また, OmniEyeBall を使用する際は, 120° の間隔で, 球状ディスプレイを囲んで座っていた (ここに実験の様子を映した画像を配置する)

5.3.2 実験項目

実験では, 定量的な評価と定性的な評価をそれぞれ設けた. 定量的評価の項目は以下の通り.

- プレゼンターが誰を見ていたのかをリアルタイムで記録 (OmniEyeBall 使用時は人物のクリック, 平面ディスプレイ使用時は人物を実際に見てもらうことが, 見るという動作に相当する) 前者はアプリケーションの操作ログを Tkinter で取得することで記録した. 後者はキーボードの指定のキーを押し続けることで記録した.
- 同時に, 傍聴者側はプレゼンターと目が合っていると感じる間, キーボードの指定のキーを押

SNS投稿用写真としての本館

桜並木



- 本館前には美しい桜並木がある
- 毎年花見客がやってくるほど

図 5.2 プレゼンターに用意したスライドの一部

しっぱなしにして、その操作を記録する。

- 記録から、見ているときに目が合ったと感じた割合 (真陽性率) 及び、見ていないのに目が合ったと感じた割合 (偽陽性率) を算出した。

また、定性的な評価は、被検者アンケートにより行った。アンケートの内容は SUS(System Usability Scale) に基づいた 5 段階評価項目 10 個。また独自の評価項目 5 段階評価項目 2 つ、記述式項目を 1 つ設けた。さらに、平面ディスプレイを使用するケースと OmniEyeBall を使用するケース双方が修了した後は、どちらの方が好ましいか理由付きで問う記述式項目を設けた。

5.3.3 被験者の動き

- ・ OEB 使用時と平面ディスプレイ使用時の 2 回の実験を行った (プレゼンター 1 人、質問者 3 人)
- ・ 平面ディスプレイ使用の実験の後、10 分程度の休憩時間を設け、OEB を使用する実験に移った

質問 1	このアプリケーションはしょっちゅう使いたくなるだろうと感じた.
質問 2	このアプリケーションは必要以上に複雑だと感じた.
質問 3	このアプリケーションは使いやすいと感じた.
質問 4	このアプリケーションを使用するのに専門家のサポートが必要だと感じた.
質問 5	このアプリケーションの機能はうまくまとまっていると感じた.
質問 6	このアプリケーションにはちぐはぐな部分が多くあると感じた.
質問 7	このアプリケーションの使い方は大抵の人がすぐに理解するだろうと感じた.
質問 8	このアプリケーションはとても扱いづらいと感じた.
質問 9	このアプリケーションを使いこなせると確信している.
質問 10	このアプリケーションを利用するまでに学ぶことが多いと感じた.
質問 11	対面のプレゼンテーションのように目の前に通話相手がいるように感じた.
質問 12	対面のプレゼンテーションにはない違和感を多く感じた.
記述式項目 1	(質問者側の場合) どのような時に目が合ったと感じたかお答えください.
記述式項目 2	OEB 使用、平面ディスプレイ使用、どちらのケースが好きだったかお答えください.

表 5.1 質問項目の一覧

- ・実験の開始前に、使用するアプリケーションの操作説明を行った
- ・また、5 分程度、事前準備したスライドの内容をある程度覚えてもらった (プレゼン中、出来るだけ映像の方を見てもらうため)
- ・プレゼンターは、プレゼン中、3 つ存在するプレゼン内容のチェックポイントに合わせ、ランダムな順番で 1 人ずつ、対応する被験者を見てもらった
- ・また、質問中は質問を受けている被験者の方向を見るように指示した
- ・OEB 使用時は、見る被験者の映像をクリックし続けることで、視線情報のログを取った
- ・平面ディスプレイ使用中は、被験者それぞれに対応したキー (映像に移っている左の人間から A,S,D) を押しっぱなしにしてもらうことで視線情報のログを取った
- ・一方、質問者側の被験者 3 人には、プレゼン中はプレゼンを聴くように指示した
- ・質問は、事前に準備した質問 3 つをランダムな順番で行った (プレゼンターから質問するように促される)
- ・上記の質問が終わったのちは各質問者から自由に質問してもらい、プレゼンと質問合わせて約 10 分使用した

5.4 実験結果

5.4.1 定量的評価

- ・定量的評価の結果は以下の通り

5.4.2 定性的評価

- ・定性的評価の、5 段階評価項目の結果は以下の通り (4 人の平均値を算出)

	真陽性率	偽陽性率
被検者 1	0.05	0.02
被検者 2	0.57	0.12
被検者 3	0.00	0.00

表 5.2 平面ディスプレイ使用時

	真陽性率	偽陽性率
被検者 1	0.06	0.07
被検者 2	0.87	0.28
被検者 3	0.19	0.00

表 5.3 OEB 使用時

	平面時	OEB 時
質問 1	2.75	3.5
質問 2	2.5	3.25
質問 3	3.5	3.25
質問 4	1.75	3.75
質問 5	3	3.5
質問 6	3.5	3.25
質問 7	4.75	3.75
質問 8	2.25	1.5
質問 9	3.5	3.5
質問 10	1.5	2.5
質問 11	1.75	3.25
質問 12	4	3.25

表 5.4 5 段階評価項目の結果

- ・記述式項目については 5-5 節にて言及する

5.5 実験結果から得られる知見と考察

・表 5.1,5.2 を見ると OEB 使用時，平面ディスプレイ使用時共に真陽性率（プレゼンターが見ていた時間の中で，目が合ったと感じた時間の割合）の最大値，最小値の差が非常に大きい（極端な例は平面ディスプレイ使用時の 0.00(一度もキーを押さなかった)）

・自由記述意見の 1 つ「発表者の顔の画像が暗くて小さいため、何となく正面を向いている時に目があったいそうだと感じた。曖昧であり、判断に困った」（平面ディスプレイ使用時）に見られるように，そもそも「目が合っている」という感覚自体があいまいで，人によって見られていると感じた機会が不ぞろいである原因であると考えられる。

- ・真陽性率は 3 人とも平面ディスプレイ使用時に対し，OEB 使用後の方が増加している

・サンプル数の少なさ、及び順序効果により、比較はできないが、本実験で使用したアプリケーションによって視線情報が共有されやすくなったという仮定に対し、一考の余地がある。

・一方で、視線情報を小窓の表示状態などからも判断できる OEB 使用時の方が、偽陽性率が下がると予想していたが、2 者は OEB 使用時の方が偽陽性率が増加した。特に、被検者 2 の偽陽性率は 0.16 上昇し 0.28 となっている。

・「正面に発表者がいて、さらに窓に発表者がいることに違和感を感じました。」(OEB 使用時) という意見があった。本アプリケーションは、「特定の個人を見ていない場合、全員に小窓を表示する」という仕様になっている。この仕様では、前回の人物クリックの結果、質問者側でプレゼンターの顔が正面に移動したのにもかかわらず、その上に小窓を表示され 2 つの顔が映ってしまう。そのため、見られていない時も見られていると錯覚した可能性がある。

・OEB を使用する場合において、上記以外にも小窓の表示方法に関する意見が多くみられた。

・「顔の切り抜きはもう少し下に置くことは可能か？」

・「OEB についての説明が少ないので仕様がよく分からなかった。小窓がなくなった時がよくわからなかった。小窓が唐突に消えるのに驚いた。」

・よって、小窓の表示方法に関しては再考する必要がある。

・また、全天周ビデオカメラにおいて、被写体の顔が小さく表示されるという意見も多くみられた。

・「3 人のうち誰か一人を見ることが難しいと感じた (視野に全員が収まっているので誰か一人にフォーカスして話しづらい)」

・「顔とカメラが遠い？からなのか、そもそも目が見えなかった」

・「発表者の顔の画像が暗くて小さいため、何となく正面を向いている時に目があっていそうだと感じた。曖昧であり、判断に困った。」(いずれも平面ディスプレイ使用時)

・全天周カメラ自体の視野の広さと、撮影領域上下が広がるような歪みによって、相対的に顔が小さくなるのが原因であると考えられる。

・よって、全天周カメラを用いて顔を映したコミュニケーションを行う際には、何らかの方法で顔部分をクローズアップする必要がある。

・「ディスプレイのサイズの 3 人とも常に視野の中に収まってはいるので、クリックした人が真ん中に来る必要はないかな…と感じた。画像そのものが移動するよりは、クリックしている間はカーソルの周辺に枠がでる…とかのほうが「ひとりにフォーカスしている感」があって良いかもしれないと思いました。」(OEB 使用時、プレゼンター側)

・PC を使用するプレゼンター側において、常にカメラの方を向くことで見たい人間を正面に捉えるように設計したが、常に 3 人とも視界に入っているためかえって混乱を招くことになった。

第 6 章

考察

6.1 本システムの問題点

- ・また、SUS のネガティブな評価項目が OEB 使用時で有意差は見られないものも、5 に近くなってしまっている。
- ・OEB 使用時のアンケート結果で GUI に関する意見が多く出ている。
- ・特に「仕様が分かりづらい」、「小窓の表示位置を変えたい」という趣旨の意見が多くみられた。
- ・小窓の位置のみならず、本来の全天球カメラの映像に被さって小窓が表示される仕様など、映像が複雑になってしまう状況もあり、それが混乱を招いた可能性がある。
- ・また、全天球カメラ自体の視野の広さから、パノラマ映像では顔が小さく表示されやすく、PC の画面に表示する全天周カメラの映像表示方法にも改善の余地がある。
- ・算出した偽陽性率に改善が見られなかった点から、小窓に表示する顔の表示方法自体も変更する余地がある。
- ・実験結果に関係しない事項としては、実験用アプリでは被験者の位置が変更しない制限があったが、実際は被写体は自由に動くことが出来るべきである。被写体が移動しても正しく小窓を表示できるようにする必要がある。

6.2 本システムの応用

6.2.1 対話相手表示形式

- 6-2-1 3 節では小窓の表示方法、あるいは顔自体の表示方法などを変更したアプリケーションを提案する
- まず、実験用アプリケーションと同様に、小窓に、PC 使用者と、見られている参加者の 2 人を表示する手法を提案する
- 実験用アプリケーションと異なる点は以下の通り
- 小窓の表示位置を、球体ディスプレイの赤道部分に寄せる
- 顔認識ライブラリ dlib を用いて、顔を認識した位置に対して小窓を表示する
- PC 使用者は、実験用アプリケーションの時と異なり、指定の位置ではなく可変な被検者の顔の位置をクリックすることで被験者を見ることが出来る
- 顔を中心に持ってくるのではなく、GUI の空いていた領域 (画面下部) にクリックした人物の

OEB side

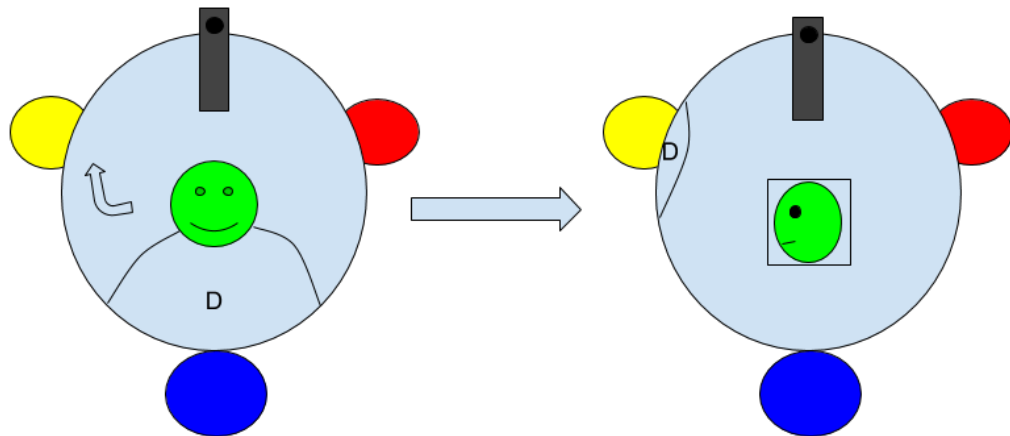


図 6.1 OmniEyeBall 上で、PC 使用者の表示位置が移動する模式図

顔の切り取りを表示する

- (dlib についての説明、アプリケーションの様子、クリックした位置に顔を持ってくるプログラムについての説明を行う)

6.2.2 横顔生成形式

- 2 人の顔をウインドウで表示する代わりに、対面で人の横顔を見るような表示
- 横顔を表示することで、立体感を拡張させ、あたかも対面しているかのような状況を作り出す
- 今回は横顔の生成は実際の映像を用いては行っていない
- Unity 上で 3D モデルを横から撮影し、UnityCapture [8] を用いて仮想カメラの映像として OpenCV で処理した
- 後述するおかしら会議 [10] のように、Unity 内の球体に画像をマッピングして、疑似的に顔を生成し、横から撮影するという方法も考えられる。
- 機械学習ベースの横顔生成手法として Yibo ら [11] の方法が挙げられる。

6.2.3 おかしら会議形式

- 全天球映像の表示方法自体を変更する手段として、宮藤ら [10] のおかしら会議が挙げられる
- 球体ディスプレイそのものを顔として見立て、全天球映像の顔部分のみをマッピングする
- マッピングする際に、全天球パノラマ画像上下部分に見られる歪みを考慮し、修正する式を用いている
- (式を乗せて、どのように歪み補正が行われているかを説明)
- (アプリケーションが動いている様子も載せる)

6.3 今後の展望

- 6-2 で、様々な顔の表示方法を提案した。
- コロナ禍の影響で十分な実験を行うことが出来なかった
- 提案したアプリケーションを用いて、十分な実験を行い、比較を行うことで、最適な表示方法を決定することが今後の課題である
- 一方で解決できていない問題も存在する
- 小窓が表示される際に、突然表示されて使用者が驚いてしまう
- 複数の人間に話しかけている場合の顔や小窓の表示方法
- 今後の研究でそれらの問題を解決する方法を模索していく
- また、今回のアプリケーションは個人対複数人といったユースケースに限られている
- 個人の参加者が複数人いた場合、OmniEyeBall 上での表示をどのようにするかという問題がある。
- 例えば、OmniEyeBall 側で、現在映したい人物を選択できるようにすれば、今回のシステムをそのまま活用できる
- 一方で、会話に PC 使用者が複数人参加している場合には、同時に映すことが出来ない
- このようなユースケースにも対応していく方法も検討する必要がある

第 7 章

結論

結論は、網羅的にかつ簡潔に。

謝辞

本論文の執筆にあたり、議論して頂いた関係者に感謝する。

参考文献

参考文献

- [1] zoom の説明
- [2] google meet の説明
- [3] The GAZE groupware system: mediating joint attention in multiparty communication and collaboration の文献
- [4] Theta V の説明
- [5] Collaboration in 360° Videochat: Challenges and Opportunities の文献
- [6] Can You See Me Now?: How Field of View Affects Collaboration in Robotic Telepresence の文献
- [7] OBS-virtual-cam の説明
- [8] UnityCapture の説明 (<https://github.com/schellingb/UnityCapture>)
- [9] OEBStudio の説明
- [10] おかしら会議の文献
- [11] Pose-Guided Photorealistic Face Rotation の文献