

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 本研究の背景	1
1.2 本研究の目的	3
1.3 本論文の構成	5
第 2 章 関連研究	6
2.1 360 度カメラに関する研究	6
2.2 全天周球状ディスプレイに関する研究	10
2.3 ビデオ会議に関する研究	14
2.4 まとめ	15
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	20
3.1 従来型ビデオ会議アプリケーションの問題点	20
3.2 提案する視線共有システム	20
3.3 技術的課題	22
第 4 章 OmniEyeBall 対 PC 間通信の実装	25
4.1 ハードウェア実装	25
4.2 ソフトウェア実装	25
第 5 章 実験	31
5.1 実験の目的	31
5.2 実験用アプリケーション	31
5.3 実験手順	31
5.3.1 実験設定	31
5.3.2 実験項目	32
5.3.3 被験者の動き	34
5.4 実験結果	35
5.4.1 定量的評価	35
5.4.2 定性的評価	36
5.5 実験結果から得られる知見と考察	38

第 6 章	考察	41
6.1	本システムの問題点	41
6.2	本システムの応用	41
6.2.1	対話相手表示形式	41
6.2.2	横顔生成形式	45
6.2.3	おかしら会議形式	48
6.3	今後の展望	52
第 7 章	結論	54
	謝辞	55
	参考文献	56
	参考文献	57

目次

1.1	zoom	2
1.2	遠隔会議の様式	2
1.3	全天球パノラマ画像	3
1.4	theta V	3
1.5	球体プロジェクター	4
2.1	画像処理の流れ [4]	7
2.2	つなぎ処理スッチング [4]	8
2.3	テイスの指示楕円 [20]	9
2.4	MeetingOWL の画面 [21]	10
2.5	iSphere [23]	11
2.6	等距離射影方式の模式図	12
2.7	正距方位図法 [20]	13
2.8	OmniEyeBall の構成の概要 [24]	14
2.9	Qoom [25]	15
2.10	Qoom の構成の概要 [25]	16
2.11	Pan らの実験の様子 [26]	16
2.12	Room2Room による通信の様子 [27]	17
2.13	t-Room を使用する様子 [27]	17
2.14	t-Room の構成 [27]	18
2.15	Kobayashi らの提案 [29]	18
2.16	Kobayashi らの提案システムのデモンストレーション [29]	19
3.1	全天周パノラマ画像のスライド	21
3.2	クリックイベント発生時の全天球ディスプレイ上の表示	22
3.3	特定の個人を見ていない場合の表示	23
3.4	映像の加工と送受信のフロー	24
4.1	OBSStudio で画面をキャプチャする様子	27
4.2	zoom での OBS-virtual-cam の映像ストリーミング	28
4.3	正距円筒図から正距方位図への変換	29
5.1	画像が移動する模式図	32

5.2	プレゼンターに用意したスライドの一部	33
5.3	傍聴者側の実験の様子	35
5.4	プレゼンター側の実験の様子	35
5.5	平面ディスプレイ使用時の 5 段階評価項目結果	37
5.6	OmniEyeBall 使用時の 5 段階評価項目結果	37
6.1	上部分が縮小して表示される様子	42
6.2	Dlib の顔検出	43
6.3	CNN の一例	44
6.4	畳み込みのイメージ	45
6.5	プーリングのイメージ	45
6.6	横顔が見える状況	46
6.7	3D モデルを撮影する様子	47
6.8	UnityCapture を用いた他アプリケーションへの映像ストリーミング	48
6.9	生成した横顔を OmniEyeBall 上で表示させた様子	49
6.10	歪みの補正前	50
6.11	歪みの補正後	51
6.12	特徴点マッチング	52

表目次

4.1	THETA S の仕様	26
4.2	THETA V の仕様	26
4.3	Glomal 350 の仕様	27
4.4	Zoom の仕様	27
4.5	OBS-virtual-cam の仕様	27
5.1	質問項目の一覧	34
5.2	平面ディスプレイ使用時	36
5.3	OEB 使用時	36
5.4	各質問の 4 人の平均値	36

第 1 章

序論

1.1 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。

90 年代から 2000 年にかけてビデオ通話や遠隔会議システムが出てきた。2020 年は、新型コロナウイルス感染症の流行により、遠隔会議の需要がさらに高まっている。遠隔会議の形態は様々であり、例えば以下のようなものが考えられる。

- 個人間を結ぶ通信
- 同じ空間にいる複数人と、遠隔地の参加者の通信
- 複数人のグループ間での通信

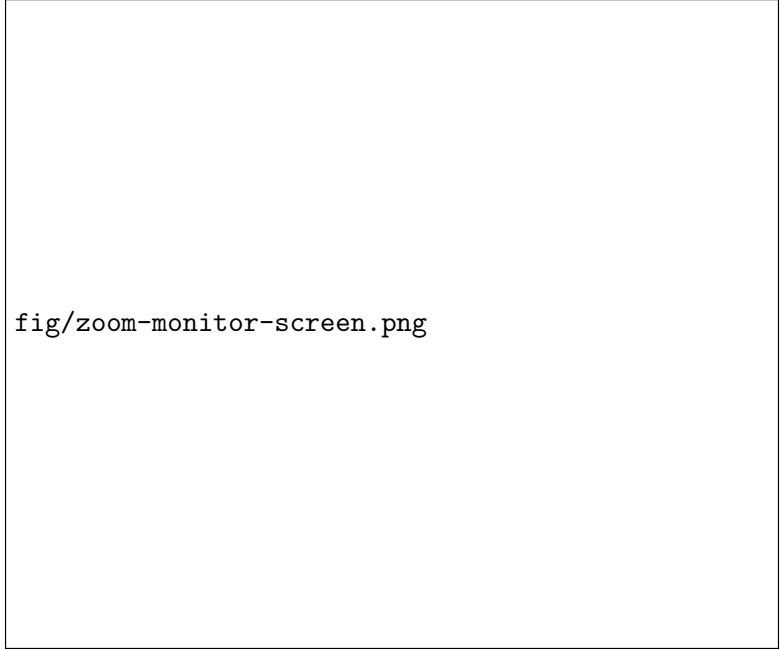
1 つ目は最も一般的な場合である。2 つ目は、例えばある病に感染するなど他の人物と接触できない人物が、グループの会話に参加する場合などに考えられる。3 つ目は、異なる研究室間のミーティング等が行われる場合である。

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。

一方で、昨今は誰でも気軽に全天球映像 (図 1.3) 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 1.4)) 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。加えて Tang ら [19] は、全天球カメラの使用者がどこを見ているかを知る手段が無ければ、コミュニケーションが困難になることを報告している。


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 [15] 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 [18] [24] がある。OmniEyeBall を用いた実験では、平面ディスプレイによるパノラマ映像の表示より、球体ディス



`fig/zoom-monitor-screen.png`


図 1.1 zoom

[1]



`fig/conference.png`

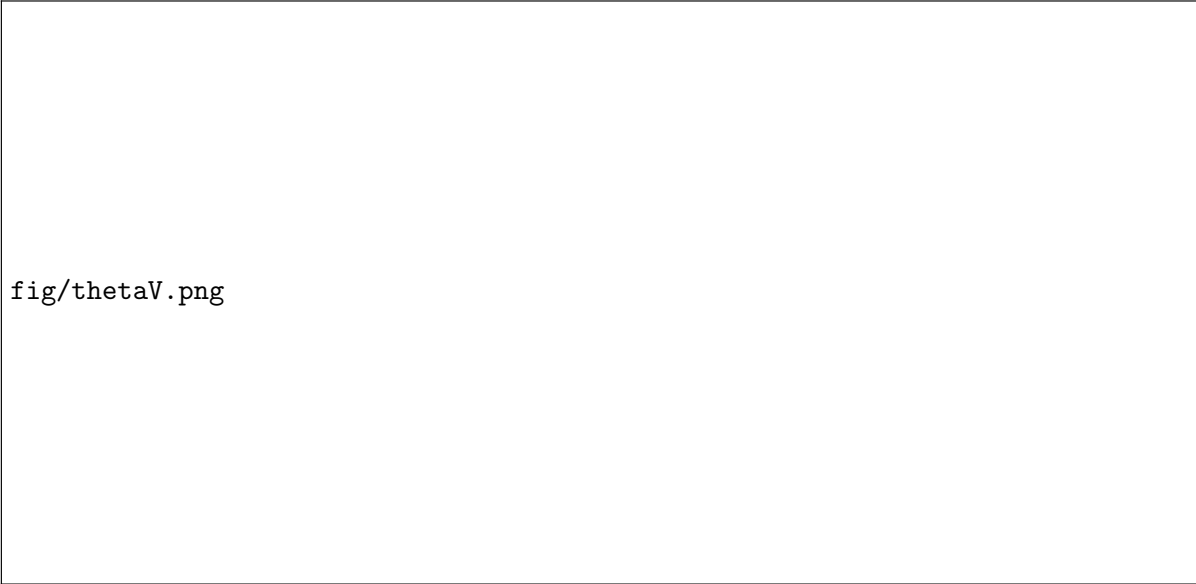
図 1.2 遠隔会議の様式



fig/panorama.png

図 1.3 全天球パノラマ画像

[4]



fig/thetaV.png

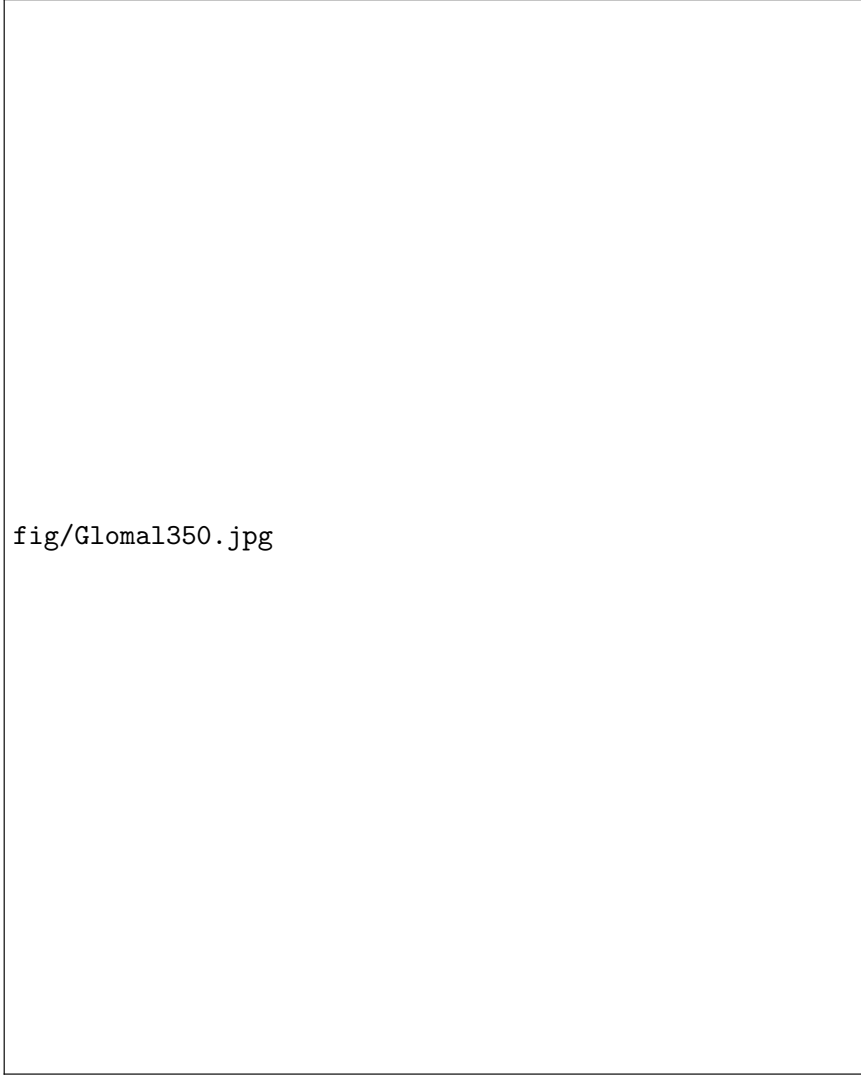
図 1.4 theta V

[4]

レイによるの表示の方が，歪みの低下，立体感の上昇などから視覚認知が向上されたことが報告されている．

1.2 本研究の目的

本研究では，個人対複数人というシチュエーションの遠隔会議において，臨場感を高めることを目的とする．この形態の遠隔会議は，従来から行われてきたものの，コロナ禍の影響において，今後さらに増えていくと予想される．



fig/Glomal350.jpg

図 1.5 球体プロジェクター

[15]

この際、参加者は全天球カメラを使用し、個人側は PC に、複数人側は OmniEyeBall をディスプレイとして使用する、非対称なビデオコミュニケーションを行うものとした。OmniEyeBall の使用理由は、立体的な表示で臨場感を高めることである。一方、個人の PC の利用理由は、入手コストの観点から、個人での利用機会が多いものを選択した。

1-1 節で述べたように、全天球カメラの使用者がどこを見ているか知る手段が必要である。特に、今回の状況では、個人側のディスプレイには複数人が表示され、通信相手に誰を見ているのか伝えるのが重要である。その方法として、映像の回転を行う。そうすることで、PC 使用者が見ている人物の方向に対して、映像に表示された PC 使用者の顔が、見られている人物の方向を向くようにする。視線情報が伝わりと同時に、顔が向くといった対面時での現象を再現することによって、会話の没入感を高めることも目指した。

上記の操作を実装したシステムを用いて、実験を行い、正確に視線情報が伝わるかを調べる。また、使用感についてのアンケートを行い、臨場感が上昇したか、及びその理由を調査する。その結果から臨場感を高める要素が何であるのかを見出し、それを反映したいくつかのアプリケーションを提

案する.

1.3 本論文の構成

本論文の構成は以下のようになっている.

第2章では, 本研究の関連研究として, 全天球カメラとそれに関する研究, 球状ディスプレイに関する研究, 及びビデオ会議に関する研究を紹介する. 第3章では, 本論文で提案するビデオ会議システムについての詳細を述べ, 第4章ではその実装を具体的に記述する. 第5章では, 本研究で行った実験について仔細を述べ, 実験結果, 及びそれに対する考察を記す. 第6章では第5章での実験結果に基づいて, 実験で用いたアプリケーションの問題点を述べる. 加えて, 作成した3つのアプリケーションについて, 詳細と具体的な実装を述べたのち, 本研究での限界と, 今後の研究課題について述べる. 第7章では, それまでに記述した内容を踏まえ, 総括と結論を述べる.

第2章

関連研究

2.1 360 度カメラに関する研究

全天球カメラ

全天球カメラの具体例としては、RICOH の THETA V [4] が挙げられる。

THETA の裏と表には、それぞれ 1 枚ずつ、計 2 枚の魚眼レンズが取り付けられている。本体を薄く設計することで、2つの魚眼レンズの視差を小さくして、後述するスティッチング処理を行った際違和感のない画像を生成することを可能にしている。

THETA のパノラマ画像は、以下のようにして生成される。まず、2 枚の魚眼レンズで撮影された映像それぞれについて、通常のカメラでも行われるような基本的な画像処理と、映像それぞれの明るさや色味が合うようにする処理を行う。生成された 2 枚の画像に対し、パターンマッチングを行い、つなぎ位置を検出する。一方で、各魚眼レンズには特有の歪みが存在し、Theta ではその歪みを補正する処理を行っている。この 2 つの操作 (つなぎ処理スティッチングと呼ぶ) を行い、2 枚の画像を結合することによって全天球パノラマ映像を生成している。(図 2.1,2.2 参照)

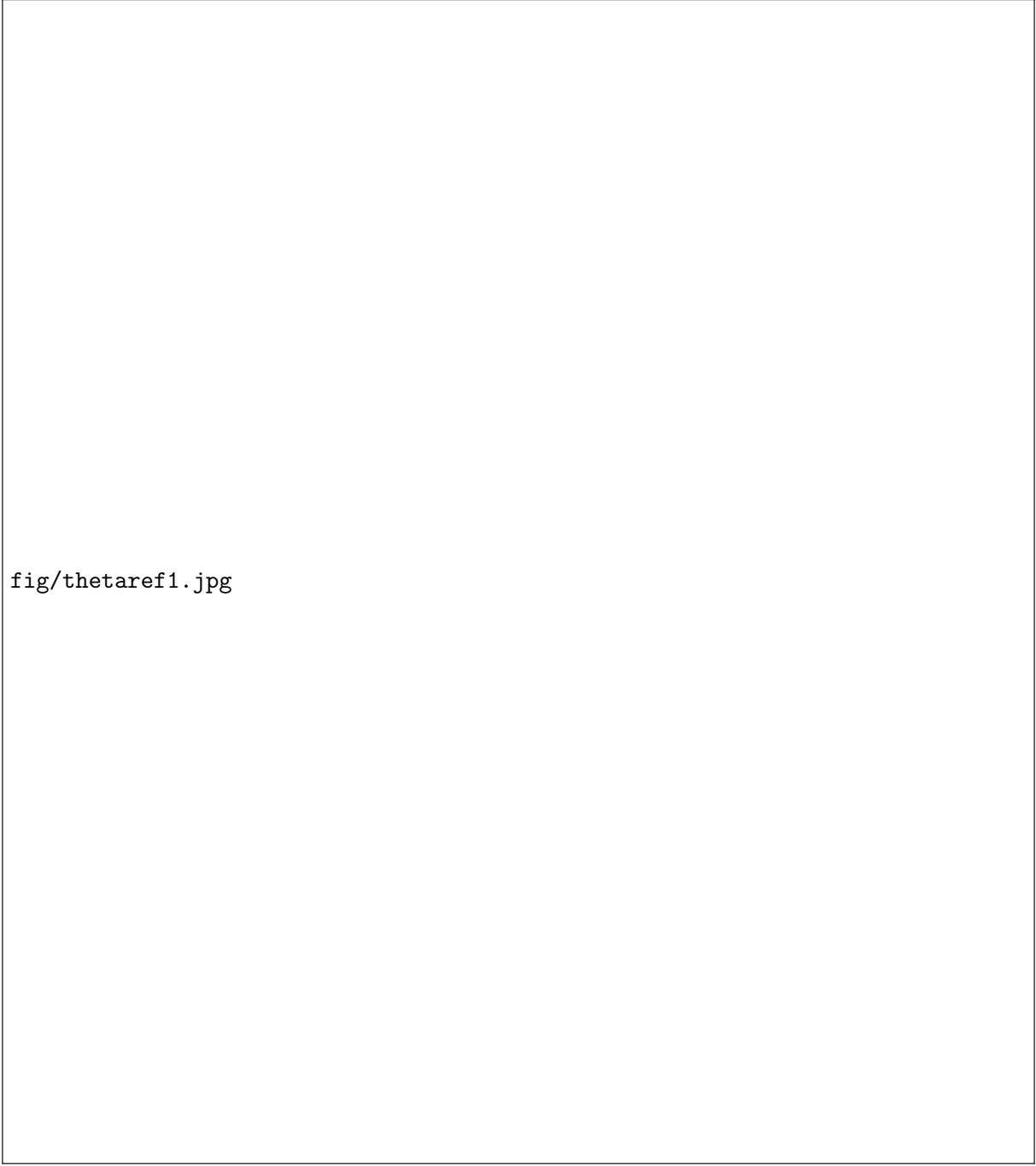
一方でパノラマ画像を生成する際に用いられる正距円筒図法であるが、これには欠点が存在する。正距円筒図法は、球面を平面に投影する方法の一種であり、緯線と経線が直行し、それぞれが等間隔になるように表示された図法であり、世界地図などに広く用いられてきた。

投影の方法上、どの経線の長さも赤道の長さと一致しているがこれは球面上の経線の性質と一致していない。実際、球面上では

$$(\text{緯度}\theta\text{の経線の長さ}) : (\text{赤道の長さ}) = \cos \theta : 1 \quad (2.1)$$

であり、正距円筒図法上では、緯度 θ 部分が横方向に $\frac{1}{\cos \theta}$ だけ拡大されていることになる。この拡大によって生じた歪みをわかりやすく示したものがテイソーの指示楕円である。(図 2.3) 高緯度領域ほど、横方向に拡大されて楕円が大きくなる様子が図示されている。

全天球カメラを利用し、さらにパノラマ画像をそのまま表示するのではなく、臨場感のあるビデオ会議を可能にしたデバイスとしては MeetingOWL [21] がある。



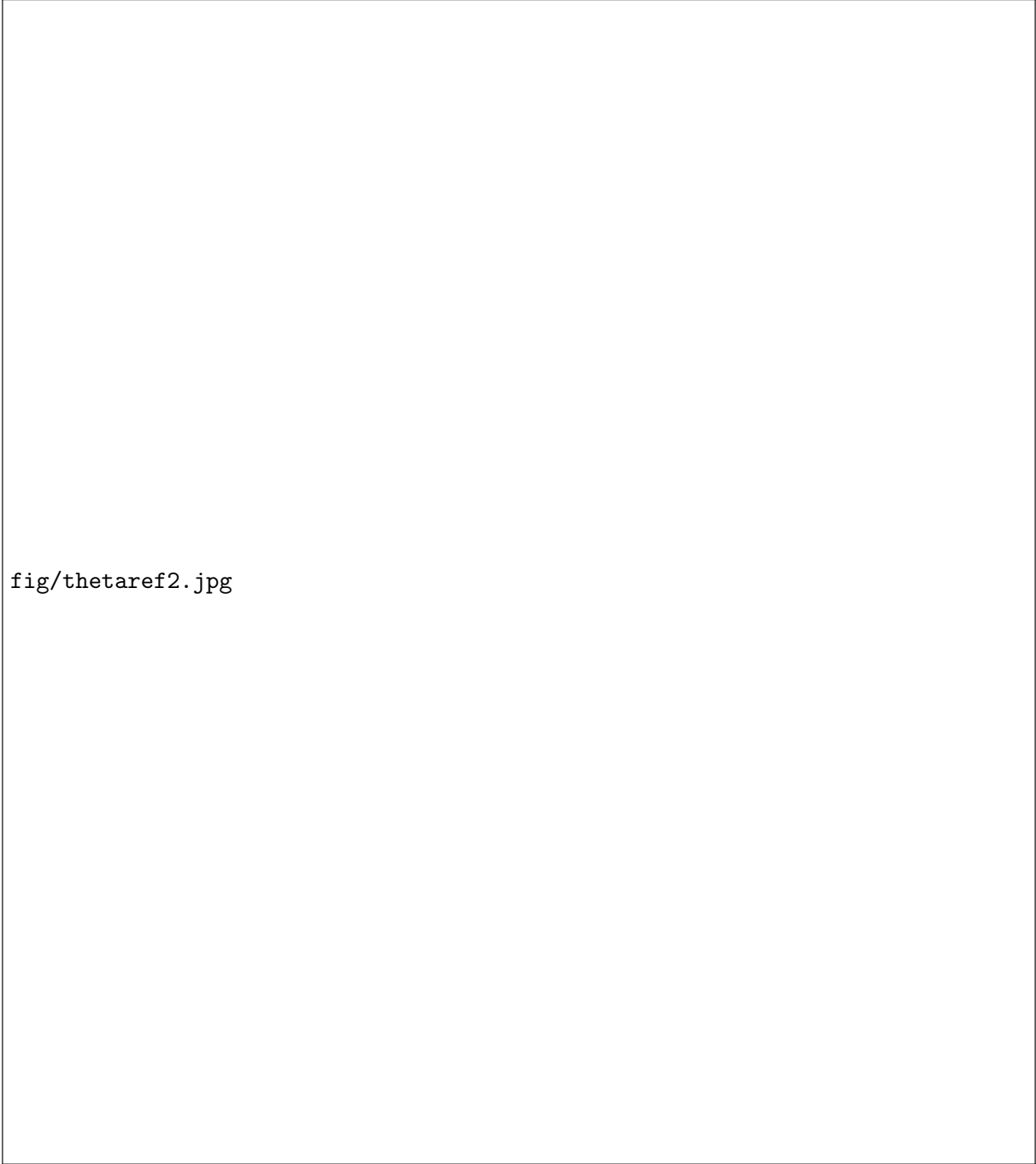
fig/thetaref1.jpg

図 2.1 画像処理の流れ [4]

全天球カメラの応用

MeetingOWL は、マイクによる声の入力と、全天球映像から得られる話者の姿勢によって、現在会話中の人物に焦点を当て、拡大して表示するシステムである。位置関係を損なわない画像の拡大により視認性が向上し、臨場感が増している。しかし、通信相手が映像に映っている人物の誰を見ているかについては、依然共有方法が存在していない。

全天球カメラを使用する効果に関する研究は、Tang ら [19] が行っている。Tang らは、360 度映



fig/thetaref2.jpg

図 2.2 つなぎ処理スティッチング [4]

像ビデオチャットアプリを作成し，それを利用して遠隔地から現地の作業相手を支援するタスクの実験を行った．その実験の結果，全天球カメラを用いることの利点として，以下を挙げている．

- 全天球カメラの視野の広さのおかげで，カメラ位置の調整の必要が少なくなり，遠隔地の参加者が自由に見たい場所を見ることができるようになった．
- 視野の広さの影響で，遠隔地の参加者がより多くの情報を獲得でき，より良く作業を支援することができた．

全天球カメラは，使用者の周囲全体を映しているので，遠隔地の参加者は使用者の周辺を自由に見

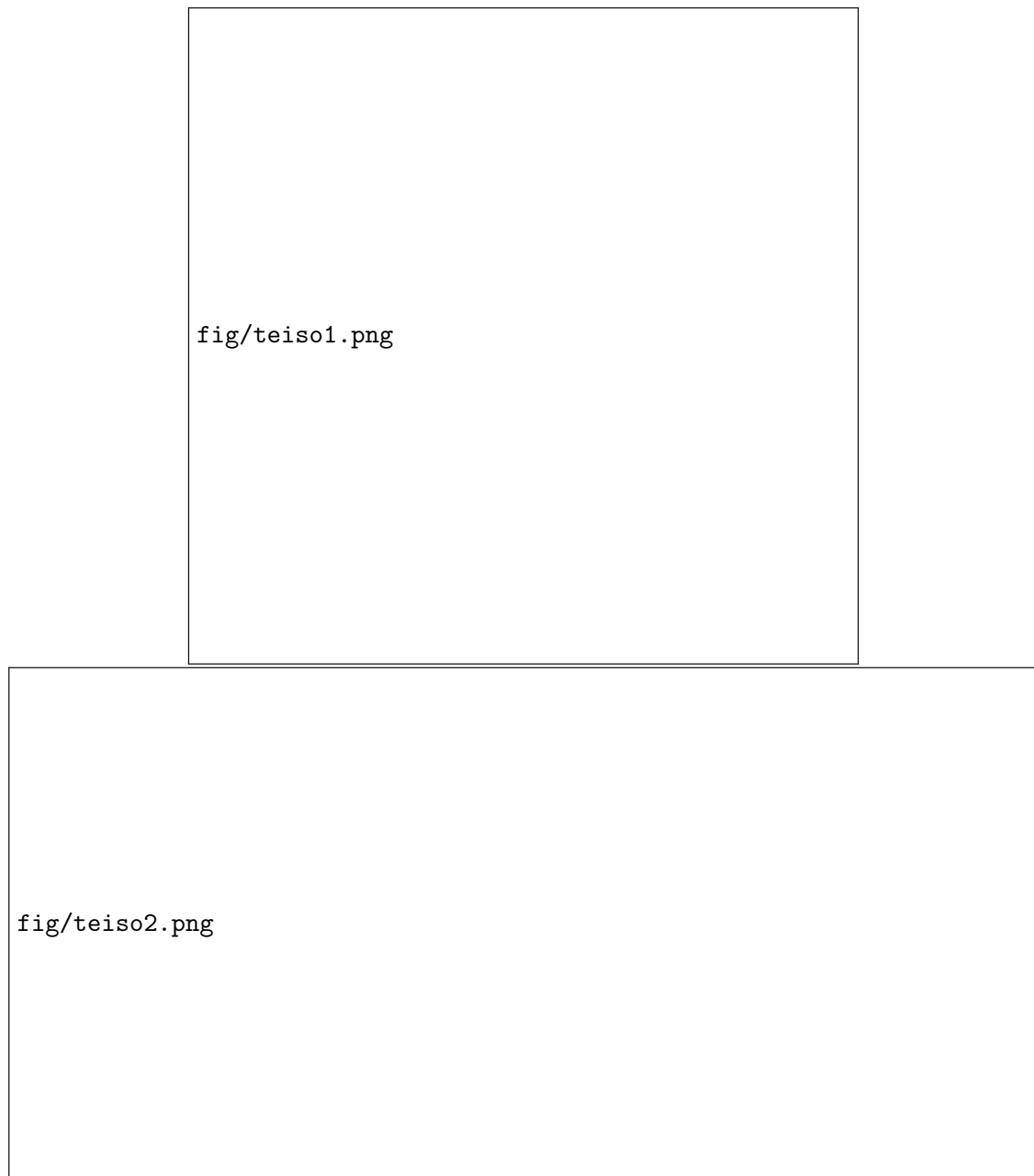


図 2.3 テイソーの指示楕円 [20]

渡せ、使用者の状況をすぐに知ることが出来る。一方で、全天球カメラの問題点として、以下を挙げている。

- 遠隔地の参加者は、自由に映像を見ることが出来る一方で、現地の参加者と見ているものが一致しない場合があった。
- カメラの角度調節などを指示する際の語彙が不足していた。(カメラを上方向に傾ける操作を「上げる」と呼び、それがカメラを上昇させる操作と誤解される等)

ここに挙げた問題点の 1 つ目は、全天球カメラの視野の広さに起因するものであり、2 つ目はコミュニケーション自体が複雑化したことに起因するものである。視線情報の共有や、3 次元的な複雑な情報の直観的な表示方法を考えることが重要である。

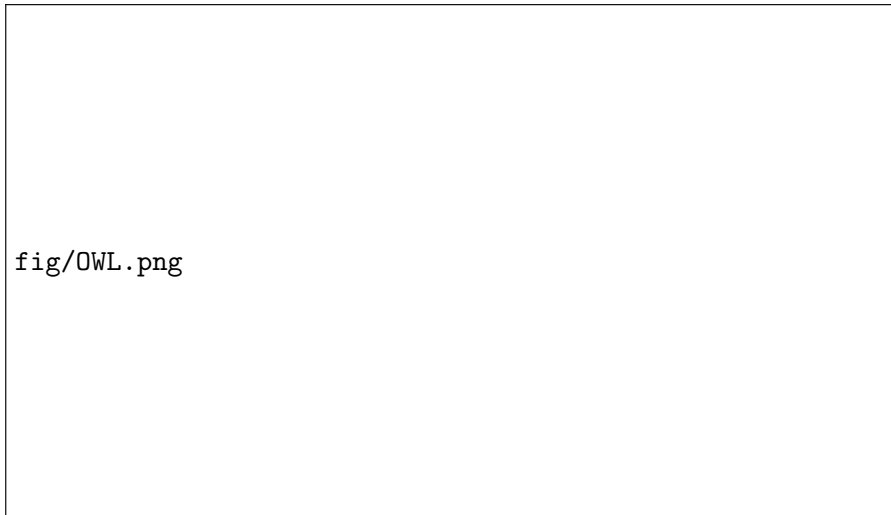


図 2.4 MeetingOWL の画面 [21]

2.2 全天周球状ディスプレイに関する研究

球体ディスプレイ

球体ディスプレイの例として、渋谷工学の Glomal350 [15] や学研の WORLDEYE [22], Wataru らの iSphere [23] がある。球体ディスプレイには

- リアプロジェクション型
- フロントプロジェクション型
- 発光型

の 3 種類が存在する。リアプロジェクション型は、ディスプレイの裏側から映像投影を行う。フロントプロジェクション型は、ディスプレイの外側から映像投影を行う。一方で、発光型は、球の構成自体を、有機 EL 等の発光装置で行い、球それ自体が発光する。Glomal550 や WORLDEYE は、リアプロジェクション型の球体ディスプレイである。一方で iSphere は、ドローンに発光 LED を取り付けて回転させて球状映像を表示する、発光型の球体ディスプレイである。(表 2.5)

リアプロジェクション型の球体ディスプレイは、球内部に魚眼レンズを取り付けており、また殆どの魚眼レンズは等距離射影方式を採用している。等距離射影方式は、レンズへの光の入射角を θ 、スクリーンへの光の到達点と光軸との距離を R 、焦点距離を f とすると以下の式が成り立つ射影方式である。(図 2.6)

$$R = f\theta \quad (2.2)$$

すなわち、投影元の画像中心からの距離に比例して、投影先の球体上での天頂からの角度が比例する。よって、魚眼レンズを使用したリアプロジェクション型の球体ディスプレイを利用するには、投影元の映像を正距方位図法 (図 2.7) へと変換しておく必要がある。



図 2.5 iSphere [23]

球体ディスプレイの応用

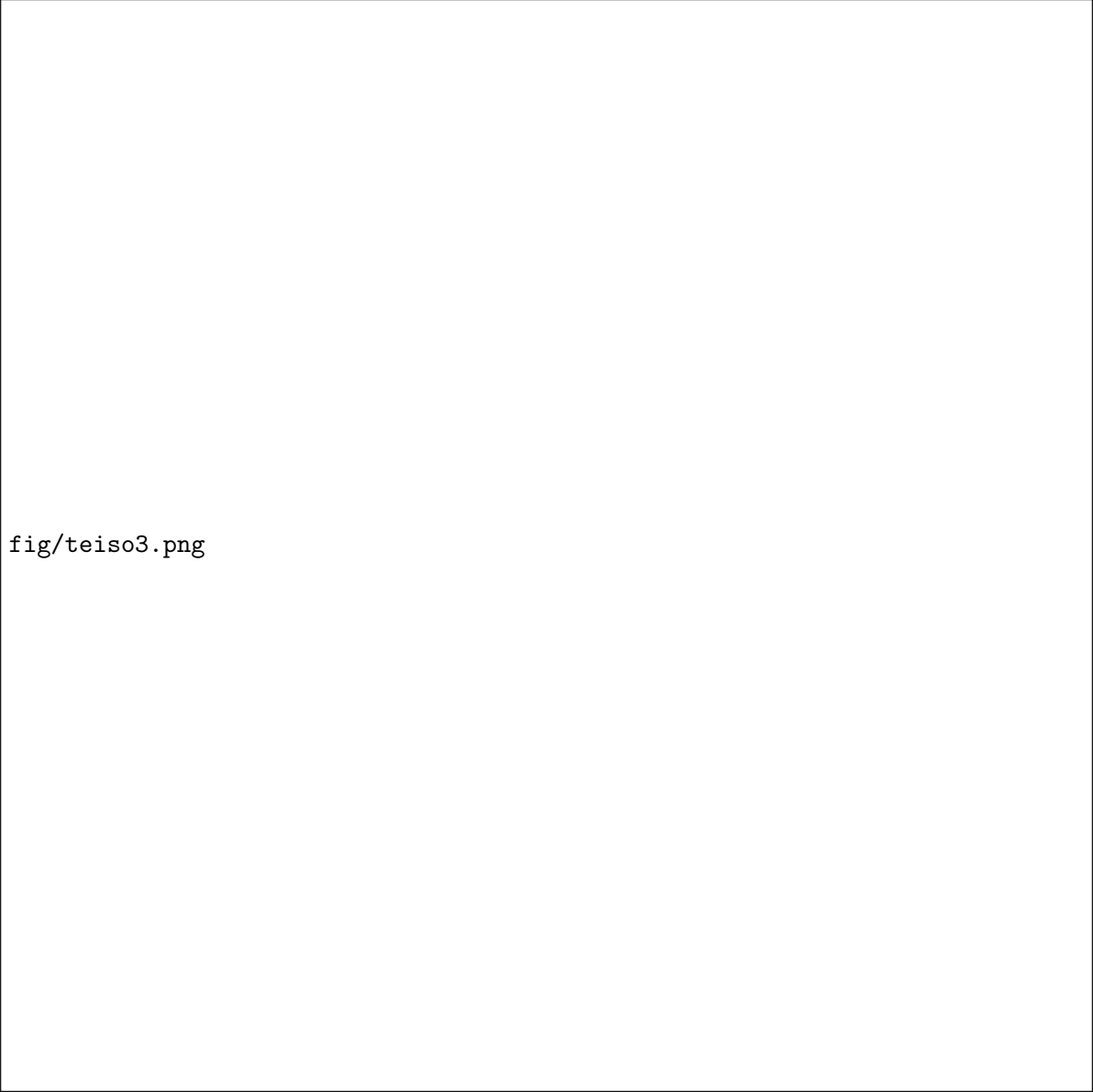
球体ディスプレイと全天球ビデオカメラを用いた例として、Li らの OmniEyeBall [18] [24] がある。OmniEyeBall は、glomal350 を映像プロジェクターとして用いて、球体ディスプレイの天頂部に広角の魚眼レンズを取り付けたデバイスである。(図 2.8)

2 台の OmniEyeBall を用いて、魚眼レンズの映像のストリーミングを行うことで、OmniEyeBall を複数人で囲う形のビデオ会議が可能である。OmniEyeBall をもちいたビデオ会議実験では、平面ディスプレイでの表示では歪みが強い、球体ディスプレイでの表示は距離や方向感覚がつかみやすいといった内容が報告されている。また球体ディスプレイを用いた方が特定の個人とのコミュニケーションがしやすくなったことが示されており、一方で平面ディスプレイ時には多対多のコミュニケーションが多く行われたことが示された。

Miyafuji ら [25] は、複数のプロジェクターとモーションキャプチャーを利用して、動くボールに対してプロジェクションマッピングを行い、ボールをフロントプロジェクション型の球体ディスプレイとして利用する Qoom を提案した。Qoom では、ボールに対して触れる、回す、地面に投げる、人に投げるといった、人がボールに対して行う自然な動作に対して、それぞれに対応した挙動をするように設計されている。

fig/projection2.pdf

図 2.6 等距離射影方式の模式図



fig/teiso3.png

図 2.7 正距方位図法 [20]

Pan ら [26] は、平面ディスプレイと球体ディスプレイでアバターの表示に関する比較の研究を行った。実験の内容は、30 個の難解な質問に対し、声と容姿は同一で正答率の異なるアバターを、平面ディスプレイと球体ディスプレイのそれぞれに表示し、どちらか片方のみから答えを聞くというものであった。(図 2.11) 結果、被検者は球体ディスプレイのアバターを多く選択する結果が得られている。特に、ディスプレイの距離の差が広いほど、球体ディスプレイを選択する傾向にあったことが示された。被検者のアンケートでは、アバターの動きや声の演者は同一であるのにもかかわらず

- エマの目はサポート感があるが、ケイティの声の方が説得力がある
- ケイティはいつも答えに自信があるように見えるが、エマは自分の知っていることを話しているように見える

などの意見が見られていた。

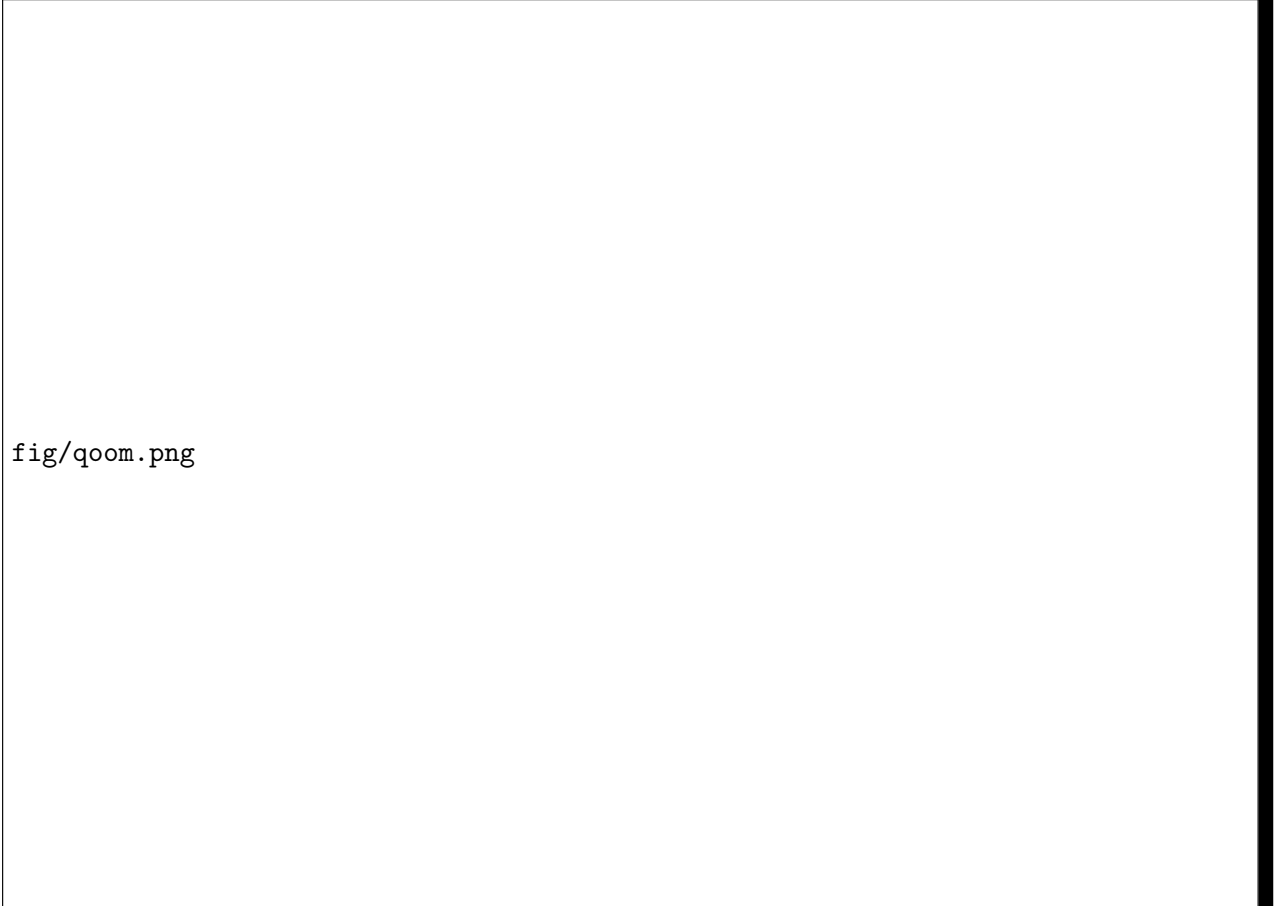


図 2.8 OmniEyeBall の構成の概要 [24]

2.3 ビデオ会議に関する研究

Pejsa ら [27] は、投影型の拡張現実環境を用いて、対面時のような 1 対 1 の遠隔会議が出来るシステムを考案した。Kinnect によって参加者の動きや体勢を認識して、3D モデルを自動生成して、相手の正面へと表示することで、疑似的な対面環境を作り出した。実験の結果、従来のビデオ会議を用いたコミュニケーションよりも通信相手の存在感や、コミュニケーションの効率といった観点で優れていることが示された。一方で、解像度の低さなどの問題から、ジェスチャーを用いたコミュニケーションの頻度が低下したことや、1 対 1 のケースに用途が限られている等の問題点が指摘されている。

Yamashita ら [28] は、卓上での作業における遠隔ビデオ会議において、ジェスチャーの情報が失われることを指摘し、卓上において遠隔参加者のジェスチャーが視認できるシステム t-Room を提案した。卓上ディスプレイに遠隔地参加者のジェスチャーを表示する際には、物体にさえぎられるなどの問題がある。そこで、リモートラグと呼ばれる、遠隔地参加者の遅延した映像をリアルタイム映像と同時に表示することで、ジェスチャーの理解の正誤判断を行えるようにした。通常表示条件とリモートラグ表示条件での比較実験の結果、ユーザーは見落とししたジェスチャーの情報などを後から再認識することに成功し、ジェスチャーを見落とすことによる不要な会話が減少したことが示されている。また、作業参加者の体感作業負荷を減少させたことも示されている。




fig/qoom.png

図 2.9 Qoom [25]

Kobayashi ら [29] は, Kinect と, 複数のカメラから視線方向の推定を行い, リアルタイムで, 対面時の会話と同じ景色を再現する映像表示方法を提案した. (図 2.15) Kobayashi らは, 2 対 2 でのコミュニケーションを行うケースを想定しており, 以下の場合について表示方法の切り替えを行った.

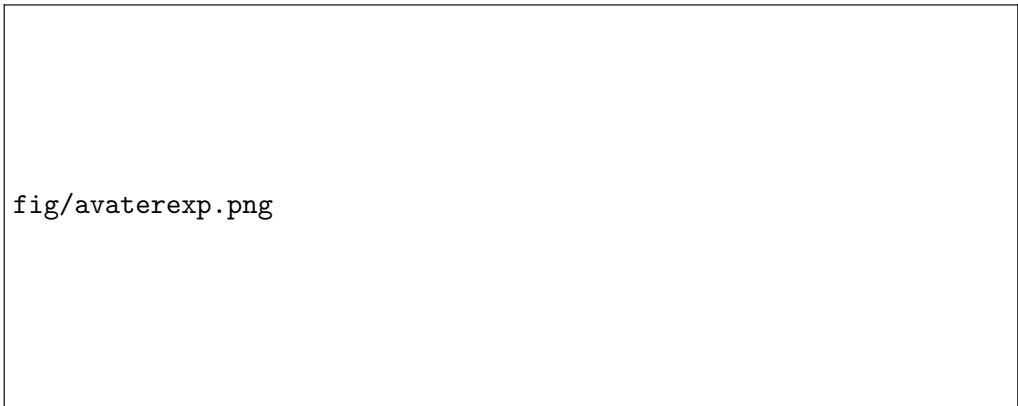
- 一方の側の人物が他方の側のユーザに見られていない場合は, デフォルトの視点位置を使用する.
- 一方の面の人物が他方のユーザからのみ見られている場合は, そのユーザの視点位置を使用する.
- 一方の面の人物が他方の面の複数のユーザから見られている場合は, お互いを見ているペアを使用する.

2.4 まとめ



fig/qoomsys.png

図 2.10 Qoom の構成の概要 [25]



fig/avaterexp.png

図 2.11 Pan らの実験の様子 [26]

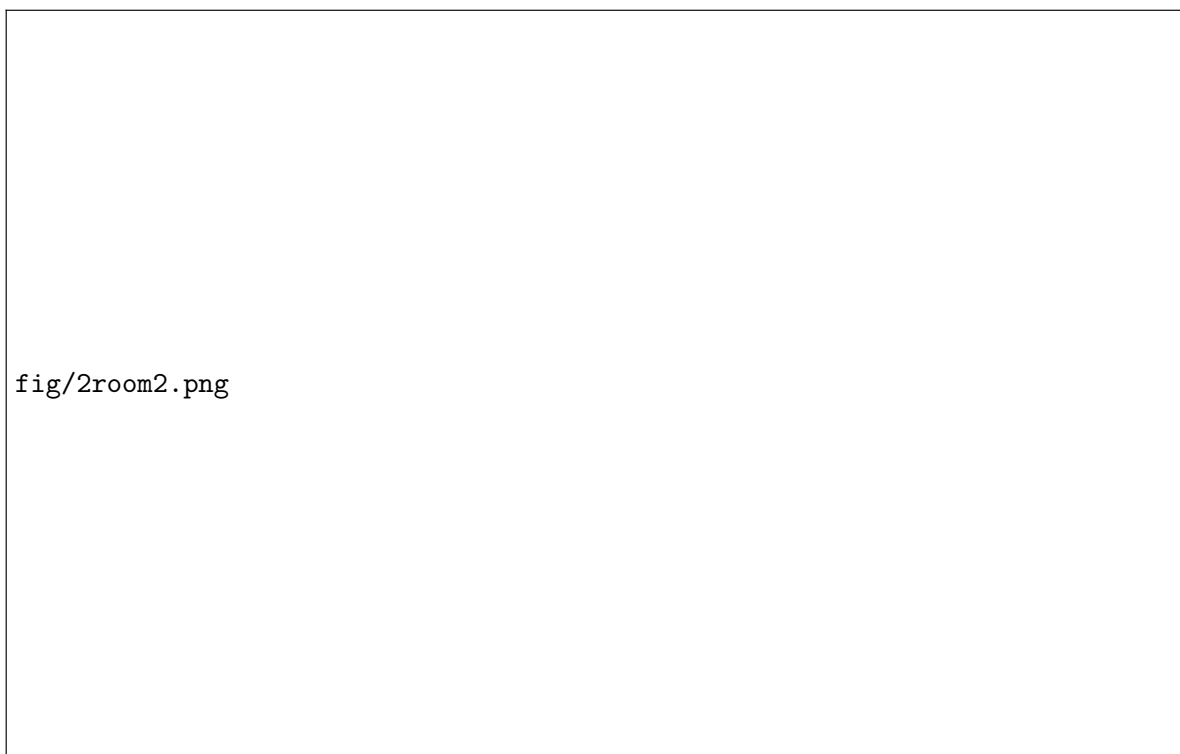


図 2.12 Room2Room による通信の様子 [27]

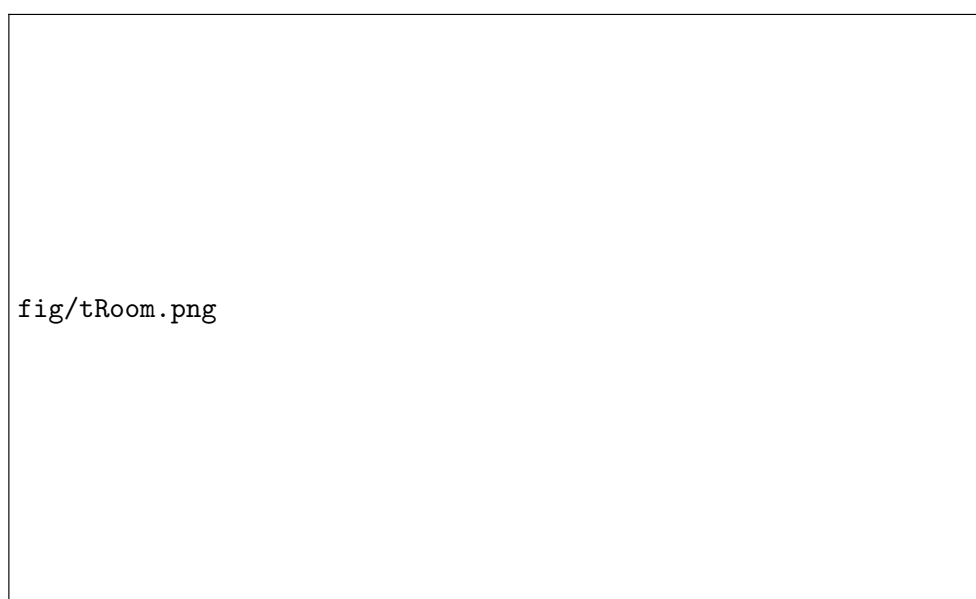


図 2.13 t-Room を使用する様子 [27]

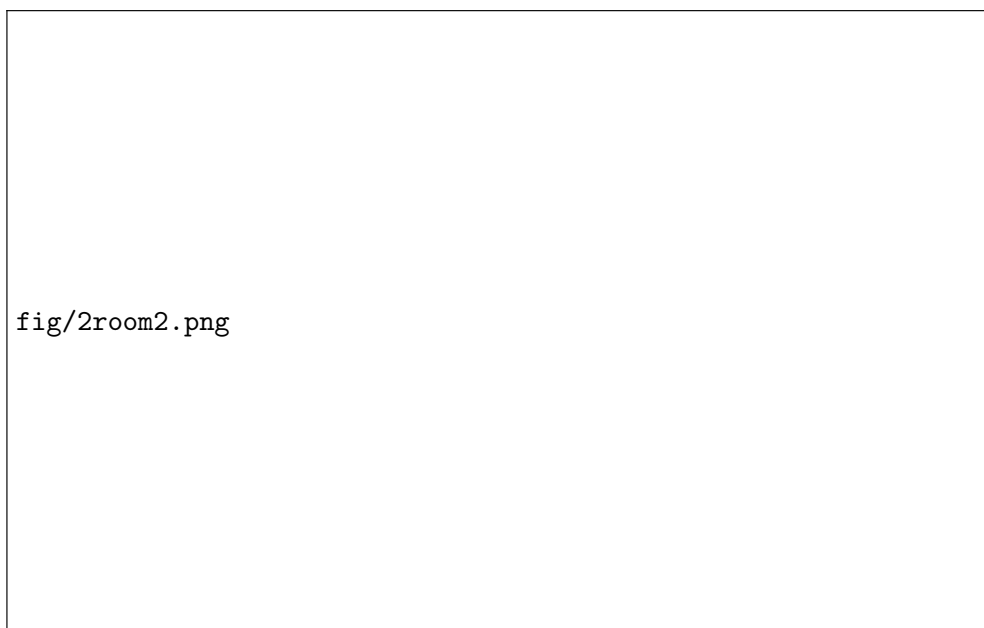


図 2.14 t-Room の構成 [27]

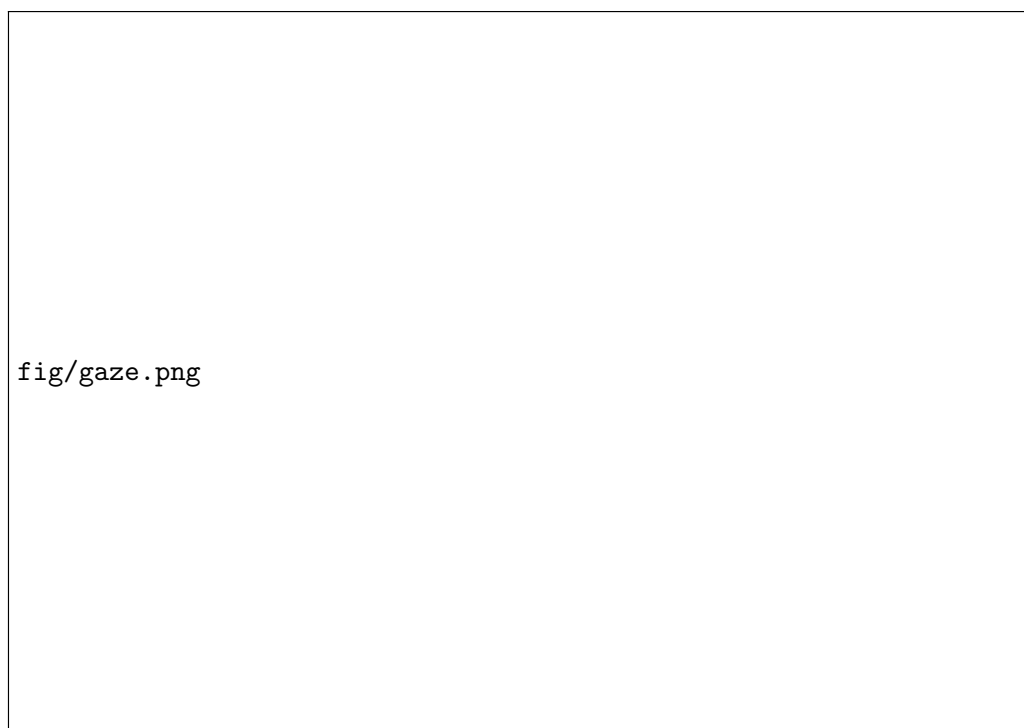
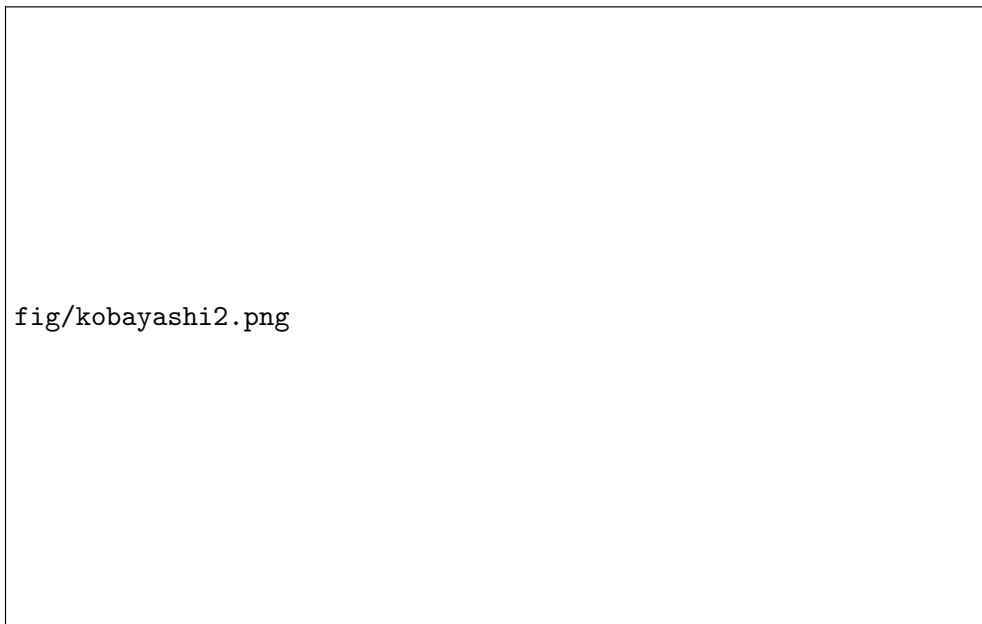


図 2.15 Kobayashi らの提案 [29]



fig/kobayashi2.png

図 2.16 Kobayashi らの提案システムのデモンストレーション [29]

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、2-2 節で述べた 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。これは、カメラ画像から発話者の視線が取得困難であったり、発話者の視線が考慮されたディスプレイ上での表示が行われていない、といったコミュニケーションにおいて重要であるにもかかわらず従来のビデオ会議では未だに未解決の問題である。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。

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。

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




図 3.1 全天周パノラマ画像のスライド

は画面上部に小さなウインドウを表示し、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 参加者の顔ウインドウの隣には、見られている参加者の顔ウインドウを表示させる。

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 参加者のみの顔ウイ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、あえて顔ウイ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。

以上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。



fig/OEsideSlideimg.png

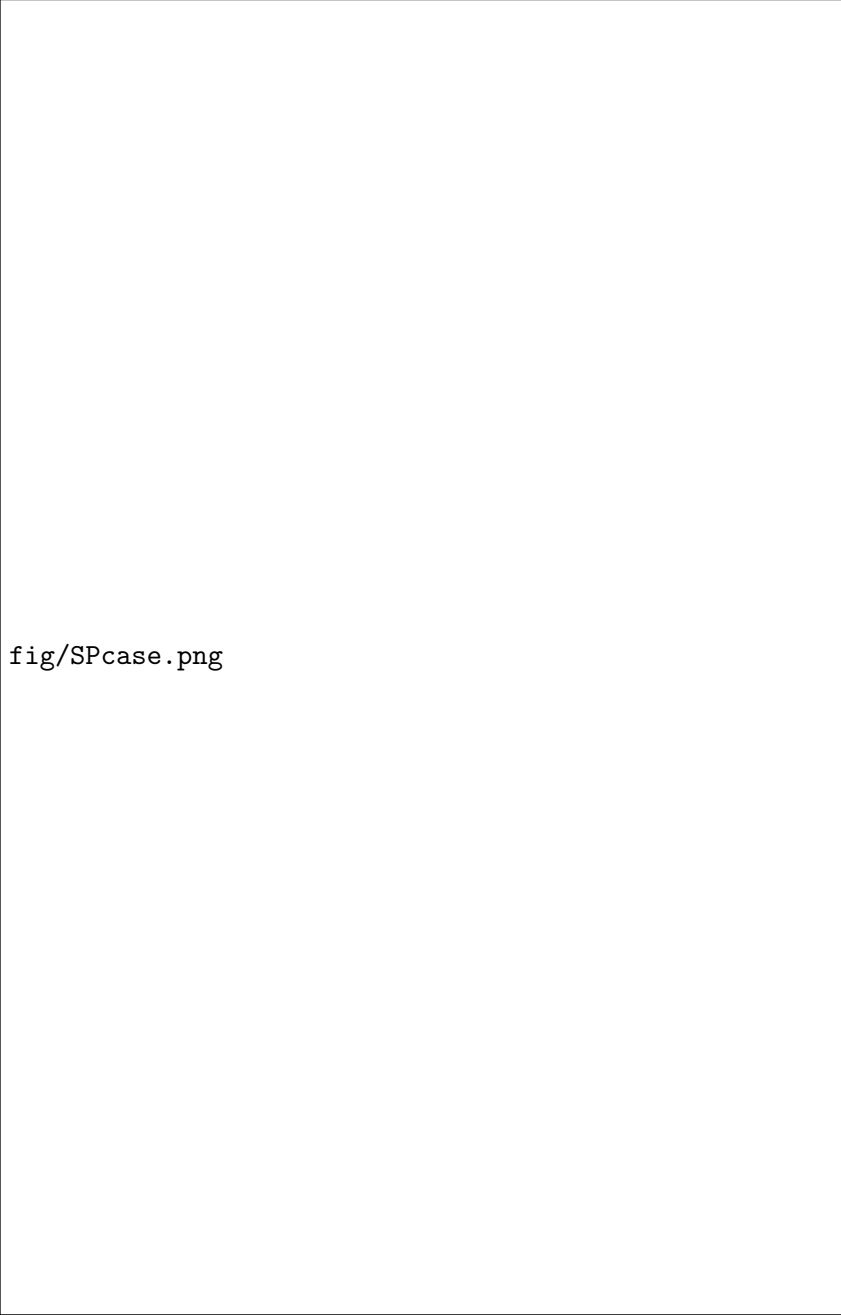
図 3.2 クリックイベント発生時の全天球ディスプレイ上の表示

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 台, 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



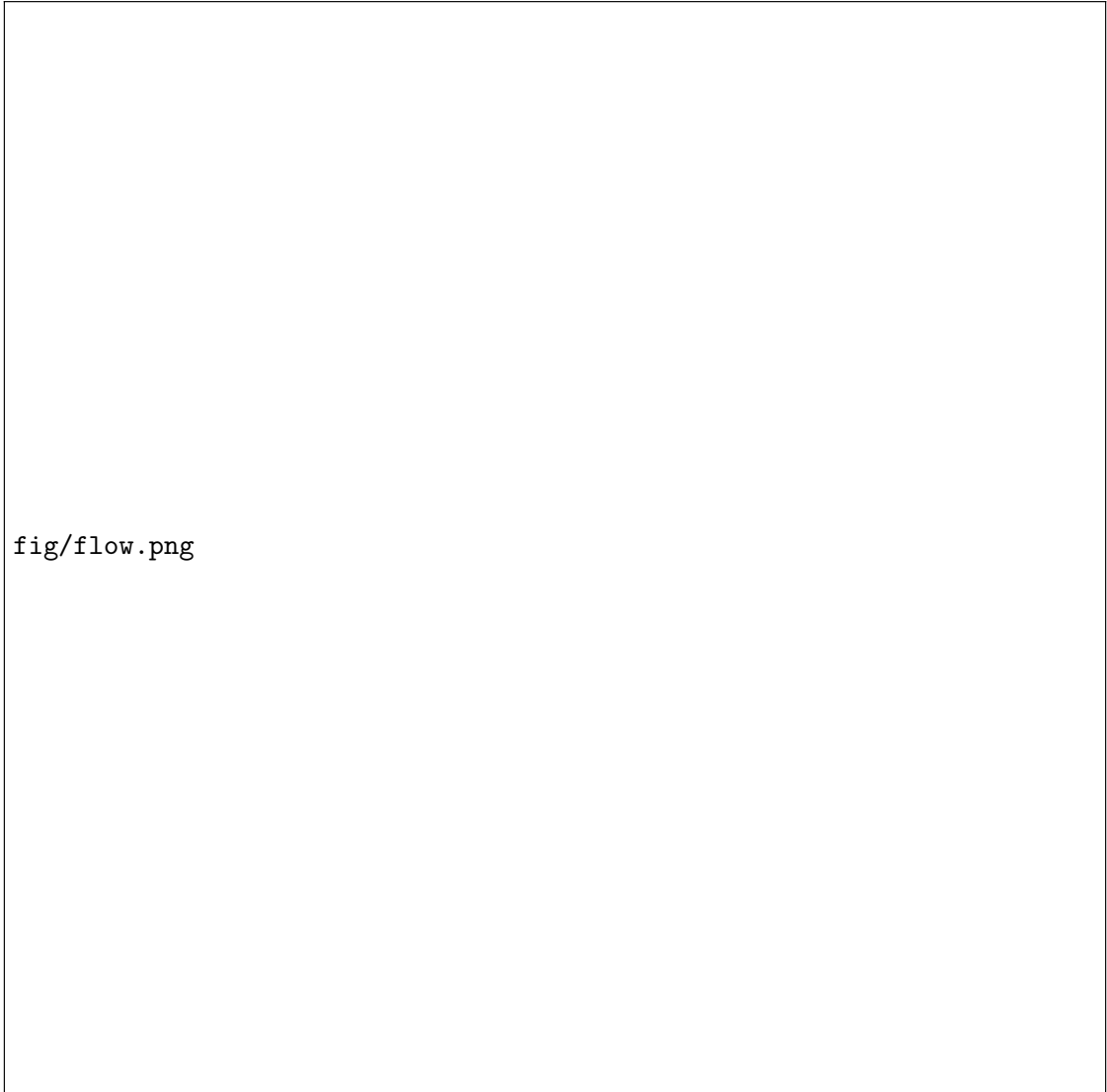
fig/SPcase.png

図 3.3 特定の個人を見ていない場合の表示

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-



fig/flow.png

図 3.4 映像の加工と送受信のフロー

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。

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. Glomal350 及び, THETA V を利用して, OmniEyeBall を設計した.

(TODO:のちに, OmniEyeBall の詳しい設計について記述する)

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.

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

記録媒体	内蔵メモリー：約 8GB
記録可能枚数, 時間	静止画：(L) 約 1600 枚, (M) 9000 枚 動画（1 回の記録時間） ：最大 25 分もしくはファイルサイズの上限 4GB 動画（合計記録時間） ：(L) 約 65 分, (M) 約 175 分
圧縮方式	静止画：JPEG (Exif Ver2.3) DCF2.0 準拠 動画：MP4（映像：MPEG-4 AVC/H.264, 音声：AAC） ライブストリーミング ：(USB)MotionJPEG, MPEG-4 AVC/H264
静止画解像度	L : 5376 × 2688 M : 2048 × 1024
動画解像度/フレームレート/ ビットレート	L : 1920 × 1080/30fps/16Mbps M : 1280 × 720/15fps/6Mbps
ライブストリーミング解像度/ フレームレート	L : 1920 × 1080/30fps M : 1280 × 720/15fps

表 4.1 THETA S の仕様

[4]

記録媒体	内蔵メモリー：約 19GB
記録可能枚数, 時間	静止画：約 4800 枚 動画（1 回の記録時間） ：最大 25 分 動画（合計記録時間） ：130 分 (2K, H264)
圧縮方式	静止画：JPEG (Exif Ver2.3) 動画：MP4（映像：MPEG4 AVC/H.264,H.265 *7、 音声：AAC-LC(モノラル)+Linear PCM (4ch 空間音声)) ライブストリーミング：(映像：H.264、音声：AAC-LC(モノラル))
静止画解像度	5376 × 2688
動画解像度/フレームレート/ ビットレート	4K, H264 : 3840 × 1920/29.97fps/56Mbps 2K, H264 : 1920 × 960/29.97fps/16Mbps
ライブストリーミング解像度/ フレームレート	4K, H264 : 3840 × 1920/29.97fps/120Mbps 2K, H264 : 1920 × 960/29.97fps/42Mbps

表 4.2 THETA V の仕様

[4]

外形寸法	幅 450X 奥行 460X 高さ 600mm 球体サイズ：φ 350mm
投影方式	1clip DLP 方式
表示素子サイズ	XGA0.55 型 (アスペクト比 4:3)
画素数	786,432 画素 (1024X768)
明るさ	2700lm
HDMI 信号	圧縮表示： 最大 1920X1080(HD TV 1080p)

表 4.3 Glomal 350 の仕様

[15]

最大解像度 1280x720

最大 fps 30

表 4.4 Zoom の仕様

fig/OBSstudio.png

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 4.5 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

なる。

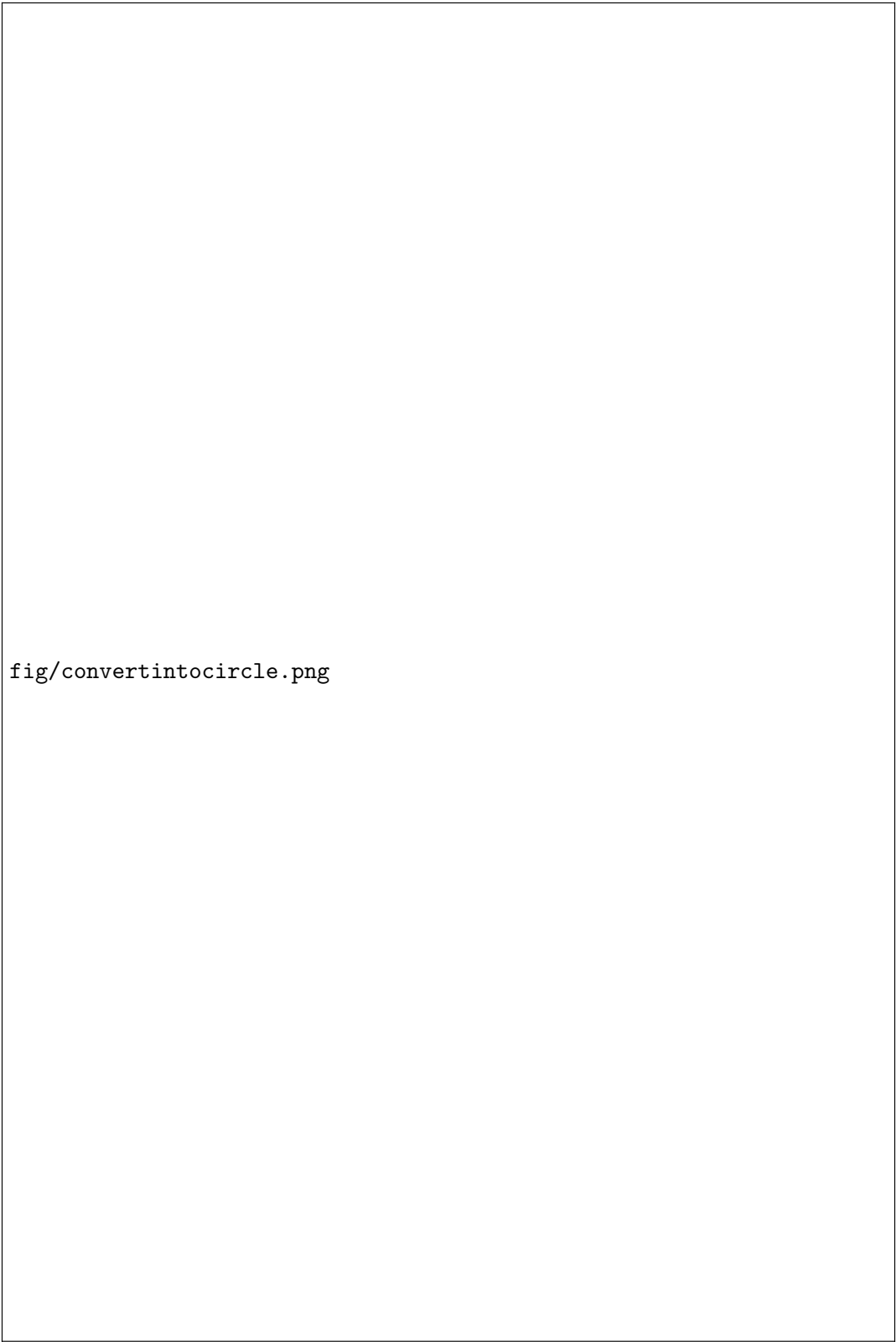
映像の加工

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 の大きさを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。



`fig/convertintocircle.png`

図 4.3 正距円筒図から正距方位図への変換

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。

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

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。

- 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それぞれ約 40cm, 50cm 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)

fig/OEBimgSlide.png

fig/PCimgSlide.png

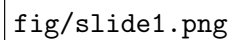
図 5.1 画像が移動する模式図

- 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

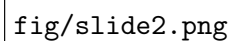
5.3.2 実験項目

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しっぱなしにして、その操作を記録する。



fig/slide1.png



fig/slide2.png

図 5.2 プレゼンターに用意したスライドの一部

- 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。

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

5.3.3 被験者の動き

この節では、実験中の被験者の行動の仔細について述べる。先述の通り、実験は OmniEyeBall 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。プレゼンターは 1 人で、傍聴者は 3 人であった。両サイドは異なる部屋で実験を行い、無線遠隔通信でのビデオ会議を行った。2 実験の間には約 10 分の休憩を設けた

実験の開始前に、使用するアプリケーションの説明を行った。プレゼンターには、各傍聴者の周辺をクリックすると傍聴者が正面にくること。及び、OmniEyeBall でどのように映像が表示されるかを伝えた。一方で傍聴者側には、自身が見られていると全天球カメラに映ったプレゼンターが個人の方向を向くこと、それ以外の状況では画面上部に小窓が表示されることを伝えた。

一方で、プレゼンテーションや質問についても一定の指示を行った。プレゼンターに対しては、プレゼンテーション前に、出来るだけカメラの方向を見てもらえるように、事前準備したスライドの内容を 5 分ほど眺めてもらい、大筋を把握するように指示した。また、準備した資料には 3 つのチェックポイントを設けて、その部分を読み上げるタイミングで、ランダムな順番で 1 人毎、傍聴者を見るように指示した。質問中は、質問を受けている傍聴者の方向を見るように指示した。さらに、誰を見ていたかをリアルタイムで記録するために追加の操作を行ってもらった。OmniEyeBall 使用時は、見ている傍聴者の映像をクリックし続けるように、平面ディスプレイ使用時は、傍聴者それぞれに対応したキーボードのキーを押し続けてもらうことで視線情報の履歴を録った。

傍聴者側には、タブレット端末を渡し、プレゼンターの紹介するプレゼン資料を事前に配布し、プレゼンテーション中にいつでも参照できるようにした。プレゼンターのプレゼンテーション中には、資料を好きに参照しながら傍聴してもらうように指示した。質疑応答の時間には、事前準備した質問 3 つをランダムな順番で 1 人 1 つずつ問うようにしてもらい、その後は自由に質問するように指示し

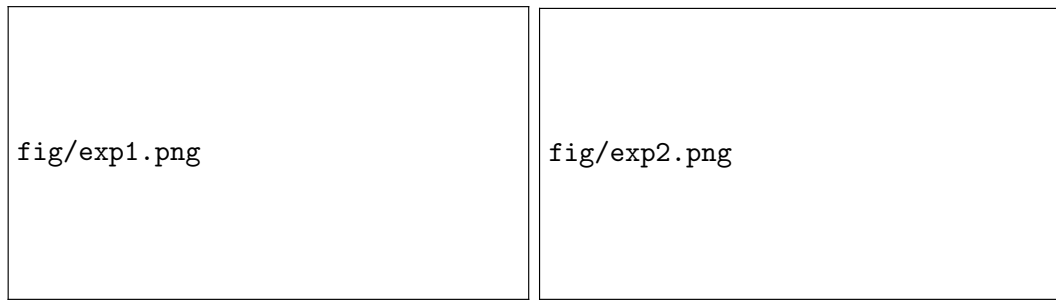


図 5.3 傍聴者側の実験の様子



図 5.4 プレゼンター側の実験の様子

た. 定量的評価のため, プレゼンターを目が合ったと感じている間は, 指定のキーを押し続けてもらうように指示した.

5.4 実験結果

5.4.1 定量的評価

定量的評価の結果を以下に示す.

	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 各質問の 4 人の平均値

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。記載した箱ひげ図については、左から質問 1 から 12 の順に並んでいる。

記述式項目については 5-5 節にて言及する。

また、平面ディスプレイを使用したケースと OmniEyeBall を使用したケースのどちらが好ましいかを問う質問では、前者が好ましいと答えた人数は 1 人、後者が好ましいと答えた人数は 3 人であった。

fig/boxplot11.png

図 5.5 平面ディスプレイ使用時の 5 段階評価項目結果

fig/boxplot10.png

5.5 実験結果から得られる知見と考察

定量的データの考察

目が合うという感覚の曖昧さ

5.2 と 5.3 を見ると、OmniEyeBall 使用時、平面ディスプレイ使用時共に真陽性率の最大値と最小値の差が非常に大きいことが分かる。極端な例としては平面ディスプレイ使用時に 0.00 という値が見られ、これは実験中に 1 度もキーを押さなかったことを表している。

自由意見の一つに

- 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。(平面ディスプレイ使用時)

というものが見られた。今回、各個人の「目が合っている」という感覚に重点を置き、あえてそのままの形で計測を行ったが、この結果にあらためて「目が合っている」と感じる状況そのものが個人によって如何に異なるかが表れている。

真陽性率と偽陽性率について

真陽性率については、3 人とも平面ディスプレイ使用時に対し、OmniEyeBall 使用時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、比較は出来ない。しかし、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ことは一考の余地がある。

- 明示的に「話しかける対象を指定する」という動作をする必要があるので、モニターに複数人が映されているという話しかける相手を意識しづらい状況で、より今誰に話かけているかを強く意識できたのが良かった

というようにプレゼンターからのコメントも見られ、視線それ自体を意識する機会を向上させた可能性もある。

一方で、OmniEyeBall 使用時の偽陽性率も、平面ディスプレイ使用時のそれ以上となっている。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。偽陽性率は、見られていないのに見られていると錯覚した時間の割合であり、視線情報が正しく伝わっていないことを示す。自身が見られている場合と見られていない場合で表示方法が異なる今回の仕様では予期していない結果であった。この原因としては「特定の個人を見ていない場合、全員に小窓を表示する」という仕様にあると考える。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。全天球カメラの映像のままでは、見られていると錯覚するため、あえて小窓を上被せた。しかし、自由記述欄には以下のコメントが存在した。

- 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。(OmniEyeBall 使用時)

加えて

- 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。(OmniEyeBall 使用時)

このように、小窓の表示方法が複数存在し、かつ直感的でなかったことが原因であったと考えられる。

定性的データの考察

5 段階評価項目は、奇数番目の項目が肯定的な、偶数番目の項目が否定的な項目となっている。サンプル数は 4 であり、比較は出来ない。(実際に各質問事項に対し、対になっているデータの t 検定を有意水準 5% で行ったところ、p 値の最小は質問 11 の 0.058 であり、有意差は見られなかった。)

だが、質問事項 4,11 のように、回答に偏りがあるものも見られ、これらから本アプリケーションの優れていた点と改良すべき点を模索する。

対面感についての考察

質問 11 の結果に注目すると、平面ディスプレイの時は 1 又は 2 であった回答が OmniEyeBall 使用時には 3 又は 4 にまで上昇している。これは、実験用アプリケーションを用いることで、ビデオ会議の参加者とあたかも対面しているかのような感覚を増させることが出来たことを示唆している。実際に、以下のようなコメントが見られた。

- 発表者の見え方が対面の時に近くて、直感的であったため。また、質問者と発表者が会話している様子がわかりやすかったから。(記述式項目 2 の回答)

会話状況の把握についての考察

5 段階評価項目としては設けていないが、記述式項目 2 の回答に会話状況が把握しやすくなったという旨の記述が 4 人中 2 人に見られた。

- 発表者の見え方が対面の時に近くて、直感的であったため。また、質問者と発表者が会話している様子がわかりやすかったから。
- 今誰に対して話しているのかが明確になってわかりやすいから。話聞いていなくて誰に対して話しているか分からない時に、即座に誰と話しているかどうか認識できるので良いと思った。

アプリケーションの仕様の複雑さの考察

質問 4 の結果を見ると、平均値に大きな差が出ており、OmniEyeBall を使用するケースの方が否定的な結果になっている。質問 4 はアプリケーションの仕様の分かりやすさについての項目である。自由記述のコメントには本アプリケーションの仕様の複雑さの指摘が 2 つ見られた。

- 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 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。

どちらも小窓の表示方法に関する指摘である。小窓の表示の改良は喫緊の課題であるといえる。

全天球カメラの視野に関する考察

最後に、記述式項目の回答に、全天球カメラの映像それ自体に関わるものがいくつか見られた。

- 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。(平面ディスプレイ使用時)
- 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(プレゼンター))

以上のように、ビデオ会議の参加者が小さく表示されてしまうという問題がある。全天球ビデオカメラの視野の広さと相対的に、被写体が小さくなってしまったり、あるいは正距円筒図法の特徴として、カメラ上・下部の映像ほど拡大されてしまうせいで、中心付近に移る顔画像が相対的に小さくなってしまふことが原因として考えられる。

本節のまとめ

以上の考察から、本システムの優れている可能性のある点として以下が挙げられる。

- 視線の情報を伝えたり、得ようとする機会が向上する
- 立体的な映像から、対面しているかのような感覚を得やすい
- 会話状況の把握がしやすくなる

一方、改善すべき点としては以下が挙げられる・

- 小窓の表示方法の改善
- PC側で表示する全天球パノラマ映像の顔部分の表示

第 6 章

考察

6.1 本システムの問題点

5-4-3 節で、立体的な映像から、対面しているかのような感覚を得やすいという結論が得られた。これに基づいて、実際に対面した時に見えるような立体視を再現することが効果的であると考えられる。

一方で、5 段階評価項目の結果では、OmniEyeBall 使用時の方が悪い結果が出ている項目も多く、実験用アプリケーションには使用感に問題があったと考えられる。自由記述アンケートでは「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ことに裏付けられている。実験用アプリケーションでは、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。

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。

以上の点を踏まえ、改善すべき点を反映させたアプリケーションを以降で提案する。

6.2 本システムの応用

6.2.1 対話相手表示形式

この節では、小窓に PC 使用者と見られている人間の 2 つの顔を表示するアプリケーションを提案する。実験用アプリケーションとの変更点は以下のとおりである。

- 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 [12] 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験者の顔の位置をクリックすることで被検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顔部分を表示する



図 6.1 上部分が縮小して表示される様子

変更点の 1 つ目は

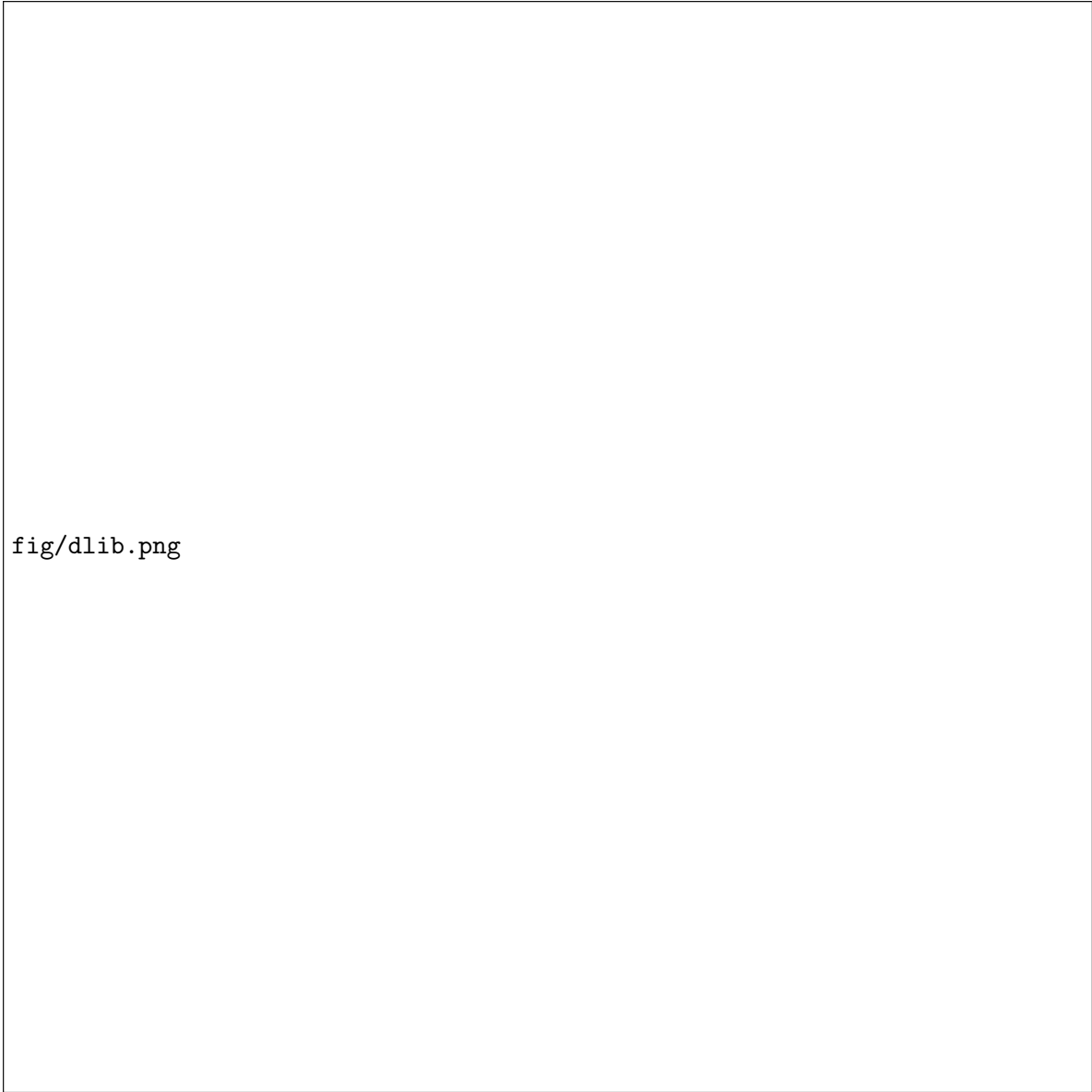
- 顔の切り抜きはもう少し下に置くことは可能か？

という、被検者からのコメントに対しての実装である。実際に、正距円筒図法によって出力された全天球パノラマ画像に、正方形で切り取った顔部分の画像を張り付けると、歪みを考慮していないために、上部分ほど実際より小さく表示されてしまうという問題がある。例えば緯度 60° に相当する部分は、正距円筒図法上では赤道と同じ長さであるが球体ディスプレイ上では赤道の長さの半分になってしまう。(表 6.2.1 参照) よって、歪みの影響の少ない球体ディスプレイの赤道近くに顔の小窓を表示することが適切であると考えた。

変更点の 4 つ目は

- ディスプレイのサイズの的に 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。
- 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)

というプレゼンターからのコメントを反映させた。また、顔にフォーカスし、表示させることで、全天球カメラの視野の広さによって顔が相対的に小さく見えてしまう問題の解決も目指した。



fig/dlib.png

図 6.2 Dlib の顔検出

[13]

Dlib

顔認識には, C++ や python で利用可能な, クロスプラットフォームソフトウェアライブラリである Dlib を用いた. Dlib には様々な種類の顔認識モデルが事前に準備されている. 代表的なものを以下に示す.

- HOG(Histograms of Oriented Gradients)+SVN(Support Vector Machine)
- CNN



fig/CNNep.png

図 6.3 CNN の一例

[13]

HOG+SVN

HOG は局所領域の輝度の勾配を求め、ヒストグラム化した特徴量である。画像はブロックと呼ばれる局所領域に分割され、さらにブロックはセルと呼ばれる局所領域に分割される。セル毎に各ピクセルの輝度勾配を計算し、勾配の方向で分類したヒストグラムを作成する。各セルの各勾配方向におけるヒストグラムは、ブロック内のヒストグラムの総和によって正規化される。

SVN は、データを 2 クラスに分類するために最適な超平面を求めるアルゴリズムである。 n 次元空間における超平面は、次元が $n - 1$ の平坦な空間であり、元の空間を 2 つに分割する。

CNN

CNN は、従来のニューラルネットワークに、畳み込み層やプーリング層といった層を組み合わせで作られるニューラルネットワークである。画像認識の分野ではよく用いられている。

畳み込み層では、学習可能なパラメータであるカーネル及びバイアスを用いて、入力に対して畳み込みが行われる。画像データの畳み込みに際しては、ある注目ピクセルとその周辺のピクセルが出力に用いられることで、画像の空間的な情報が損なわれにくいという利点がある。

カーネル K のサイズが $N * N$ であった場合、画像のあるピクセルの周辺 $K * K$ サイズの領域 A にカーネルを適用した結果は、 K と A のアダマール積となる。すなわち

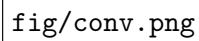
$$\sum_{i=1}^N \sum_{j=1}^N k_{ij} a_{ij}$$

である。

カーネルを適用する領域は通常 1 ピクセルずつずらされる。そうして全ての領域にカーネルが適用され、バイアスを加算した結果が畳み込み層の出力として用いられる。

プーリング層は、通常畳み込み層の直後の層であり、畳み込み層の出力を、カーネルを用いてダウンサンプリングした結果を出力する。

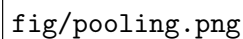
ここで用いられるカーネルは、カーネル適用範囲の平均値を出力するものと、カーネル適用範囲の



fig/conv.png

図 6.4 畳み込みのイメージ

[14]



fig/pooling.png

図 6.5 プーリングのイメージ

[14]

最大値を出力するものがよく用いられている。前者の場合は Average Pooling 層と呼ばれ、後者なら Max Pooling 層と呼ばれる。カーネルの適用範囲のずらし幅はストライドと呼ばれ、様々な値が設定されている。

今回のアプリケーションでは、より精度の高い CNN を用いたモデルを使用した。一方で、HOG+SVN を用いるモデルは、識別実行速度が速いという利点がある。

6.2.2 横顔生成形式

6-2-1 で会話中のビデオ会議参加者を表示する方法を提案した。本節では立体的な映像を表示するという部分に着目し、小窓に横顔を表示する方法を提案する。

表 6.6 のように、複数人で対面している状況を考える。話者が自身に話しかけていない場合、別の



図 6.6 横顔が見える状況

人物に顔を向けている場合が多く、その際には話者の横顔が視認できる。ここでは、小窓に話者の正面顔ではなく横顔を表示する方法を提案する。対面時の立体的情報を再現することで、ビデオ会議の参加者に更なる没入感を提供することが目的である。


実装時間の関係で、参加者の実際の顔の横顔を再現するには至らなかった。このことは今後の課題とする。代用案として、昨今様々な場所で配布されており、入手・仕様・作成が容易な 3D モデルを利用して、横顔を表示する手法を提案する。

3D モデルの撮影には Unity [16] を用いた。Unity では OpenCV 及び Dlib が利用可能で、Dlib で取得した顔のランドマーク情報から 3D モデルの体や表情を動かすことが可能である。また、Unity では自由にシーン描画カメラを配置することができる、ここでは 3D モデルの左右にカメラを配置し、リアルタイムで横顔を撮影した。Unity 内カメラの映像は UnityCapture [8] を利用することで、容易に仮想カメラの映像として利用することが出来る。

通信の高速化のため、プロセス間通信で顔の位置情報を送信し、カメラを移動させるという事は行わなかった。OmniEyeBall の使用者から見て、右側の参加者が見られているときには左から見た

fig/unitycam.png

図 6.7 3D モデルを撮影する様子



fig/unitycapture.png

図 6.8 UnityCapture を用いた他アプリケーションへの映像ストリーミング

横顔が、左側の参加者が見られているときには右から見た横顔が表示されるようにした。

今後、ビデオ会議参加者自身の顔を生成する方法としては、後述するおかしら会議 [10] の表示法を用いる手段がある。Unity 内の仮想球体に sprite として、PC 使用者の顔写真を張り付ける。そうした球体を撮影することで、疑似的な顔写真を生成する。

機械学習ベースの方法として、横顔生成手法として Yibo ら [11] の方法が挙げられる。これらの方法を用いた表示は、今後の研究課題とする。

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。これは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という



図 6.9 生成した横顔を OmniEyeBall 上で表示させた様子

手法である。


顔の切り抜きは、Dlib を用いて検出された顔バウンディングボックスによって行い、拡大したのちに楕円形にトリミングする。その画像を背景が黒い長方形画像に張り付け、正距方位図法への変換を行う。

この際、長方形にそのまま楕円顔画像を張り付けてしまうと、上下部分がが相対的に小さく描画されてしまう。よって、その歪みを打ち消す処理を行っている。


実験の考察でも述べた通り、立体的な資格情報は、ビデオ会議での疑似的な対面経験を作り出すとみられる。よって現実の顔に近い立体表現が可能なこの表示方法は効果的であると考えられる。

問題点としては、全天球画像の顔の領域が小さいことが挙げられる。小さい画像を拡大して大きな画像に張り付けるため、画質が著しく低下してしまう。

しかし、近年は 4K 画質に対応したカメラが多く存在し、全天球カメラにも 4K 画質のものが存在する。GPU の性能向上と合わせ、高解像度のカメラ映像を遅延なく送信することができれば、この

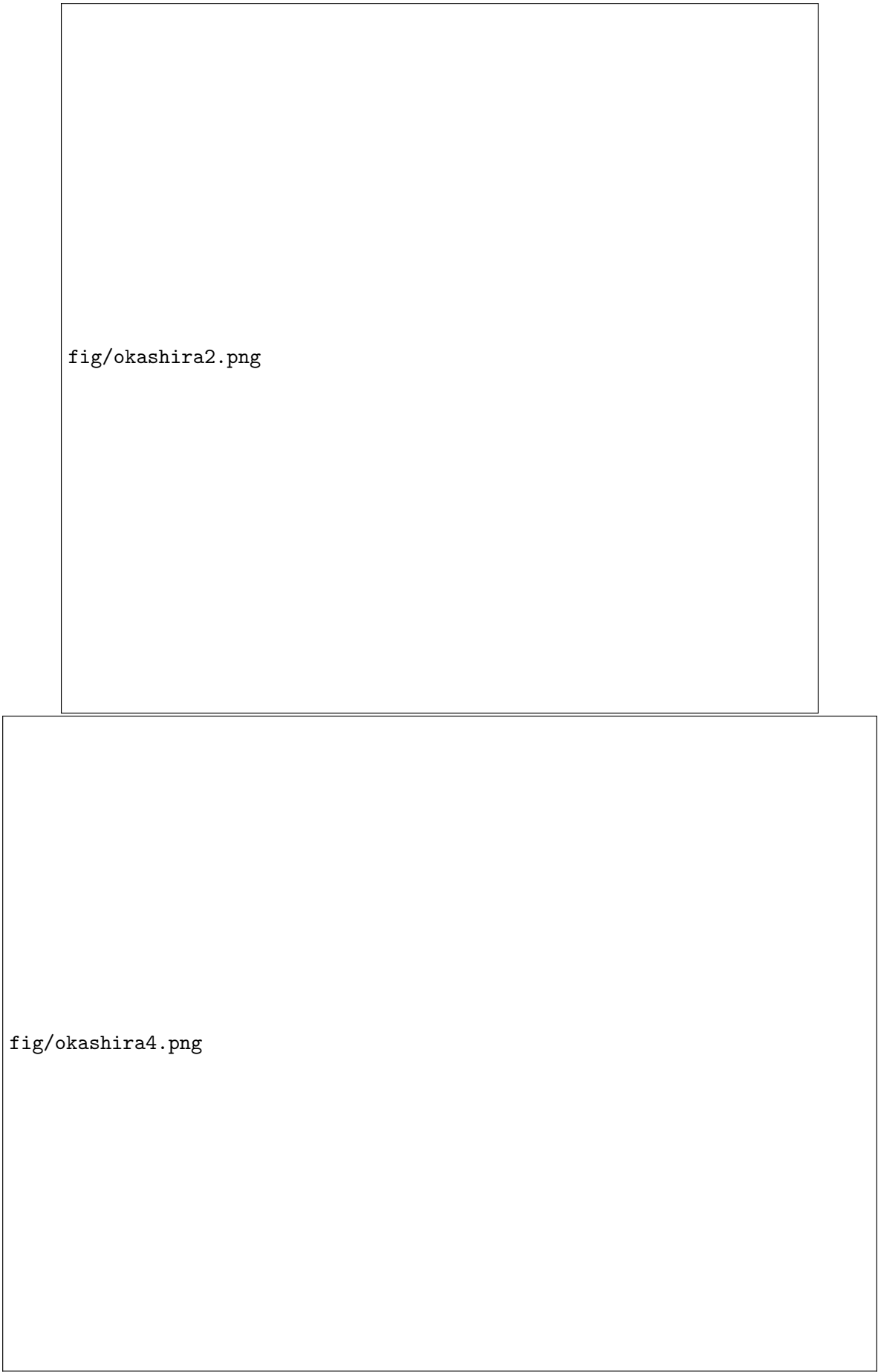


fig/okashira1.png



fig/okashira3.png

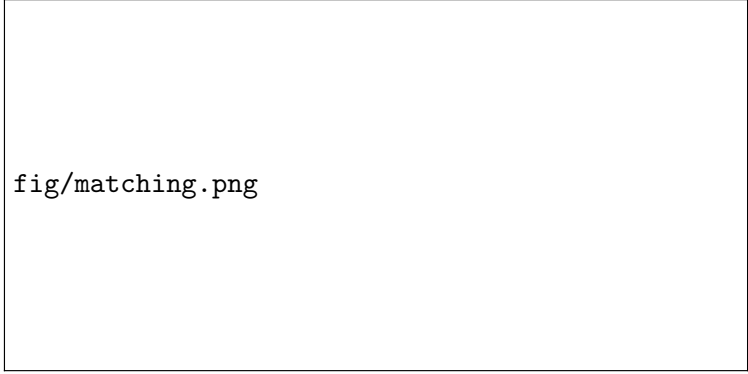
図 6.10 歪みの補正前



fig/okashira2.png

fig/okashira4.png

図 6.11 歪みの補正後



fig/matching.png

図 6.12 特徴点マッチング

[17]

問題は解決される。

6.3 今後の展望

本章では、5章で得られた知見に基づいていくつかのアプリケーションを提案してきたが、依然解決できていない問題がある。

特定の一人を見ていないケース

アンケート結果によって、小窓を既に表示されている顔の上に表示するのは混乱を招いている可能性があるという結論付けた。よって提案したアプリケーションではその表示方法を廃止した。しかし、先の表示方法は、実験用アプリケーションで特定の一人を見ていないケースを区別していた表示方法であった。この状況を表す代替の表示方法を提示する必要がある。

案の一つとしては、テキストや印を表示する方法がある。しかし、小さいとサインが見えにくく、大きすぎると映像の邪魔になってしまう。なおかつ、本アプリケーションは従来の方法に比べ、SUSの評価が向上したとは言えず、これ以上の情報提示は、ただでさえ複雑なアプリケーションをさらに複雑にすると考え、実装を見送った。

全天球ビデオカメラの回転の問題

本アプリケーションは、使用開始前の状態として、PCの使用者が全天球パノラマ画像の中心部を見ている前提で、画像のスライドや顔認識した位置に画像を張り付ける処理を行ってきた。しかし、カメラが回転すると、基準としていた位置がずれ、ずらす処理にも変更を加える必要がある。だが、本研究では回転を考慮した処理を行っていない。

これを解決する案として、特徴点マッチングを用いる方法が考えられる。特徴点マッチングによって、画像のスライド距離を算出し、その値を本論文の処理に加えて反映させれば、この問題は解決できるであろう。

様々なケースへの対応

今回は個人対複数人でのビデオ会議という状況を前提において研究を進めてきた。しかし実際には他にも以下のようなケースが存在する。

- 複数の OmniEyeBall を使用するケース
- OmniEyeBall から孤立した参加者が複数人であるケース
- 孤立した参加者の何人かが共通の空間にいるケース

例えば 2 番目のケースは、OmniEyeBall 側の参加者が現時点で注目したい遠隔参加者を 1 人選ぶというようにする。さすれば、その 1 人に対して今回提案したシステムが利用可能である。

しかし 1 番目のケース、3 番目のケースは、複数人が同じ空間を共有している。今回のような画像の回転を用いると、意図していない人物の回転を同時に招いてしまう。これらの複雑なケースに沿ったシステムを提案していくことは、今後の研究課題である。

まとめ

今後の研究方針としては、先に述べたような複雑なケースに対応するシステムの模索が考えられる。一方、今回はコロナ禍の影響で、十分に実験が行えたとは言えない。十分な時間を要して、今回提案した複数の表示方法について綿密な実験を繰り返し、より多くのユーザーの実体験に基づいた知見を獲得していくことも必要であろう。

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 (<https://theta360.com/ja/about/theta/>)
- [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
- [12] Dlib の説明 (<http://dlib.net/>)
- [13] <https://ci.nii.ac.jp/naid/40021907210>
- [14]
- [15] glomal 350 の説明 (<https://www.aisan.co.jp/products/glomal350.html>)
- [16] Unity <https://unity.com/ja>
- [17] 有志による OpenCV の解説 (http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_feature2d/py_matcher/py_matcher.html)
- [18] OmniEyeBall(<https://dl.acm.org/doi/10.1145/3266037.3266092>)
- [19] Collaboration in 360° Videochat: Challenges and Opportunities
- [20]
- [21] Meeting OWL(<http://meetingowl.jp/?i=nav>)
- [22] WORLDEYE
- [23] iSphere: Self-Luminous Spherical Drone Display
- [24] How Display Shapes Affect 360-Degree Panoramic Video Communication
- [25] An Interactive Omnidirectional Ball Display
- [26] Comparing flat and spherical displays in a trust scenario in avatar-mediated interaction
- [27] Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment
- [28] Improving Visibility of Remote Gestures in Distributed Tabletop Collaboration

- [29] A Gaze-preserving Group Video Conference System using Screen-embedded Cameras (後で正式な書式に書き直します)