

ハンドトラッキングを利用した遠隔作業支援システム

A Remote Work Support System by Hand Tracking

武政 実玖†

床井 浩平‡

Miku Takemasa

Kohei Tokoi

概要：本研究では、遠隔地における作業支援のための遠隔コミュニケーションシステムを開発した。このシステムはスマートフォンやタブレット端末を用いて、一方は手のトラッキング情報を、もう一方はリアルタイム映像を互いに送信し、映像に仮想の手を投影したAR動画の作成と共有を行う。これにより、作業支援において手の動きを離れた人に見せることができ、言葉では伝えにくい指示を直感的かつ円滑に受け取ることが可能になると考え、評価実験を行った。その結果、本システムでは直感的な操作は実現できなかったが、より円滑な指示が可能となった。

キーワード： AR, 作業支援, 遠隔コミュニケーション, ハンドトラッキング

1. はじめに

1.1 研究の背景

1.1.1 通信技術の発達

通信技術が発達し、世界中の遠く離れた人々がリアルタイムで繋がることは当たり前となっている。100年以上前に電話が発明されてから携帯電話へ、そしてここ数十年でテレビ電話へと発展した。そして、サービスを開始し始めた5GやIoTなど今後より一層通信技術が発展していくことは必至といえる。

2020年に新型コロナウイルスが蔓延したコロナ禍において、自宅からビデオ通話で会議に参加するなど自宅仕事をするテレワークは急速に広まった。図1はHR総研による今後テレワーク実施予定であるかのアンケート結果である[1]。これによると、テレワーク実施企業の約96%で今後も続けるとしている。

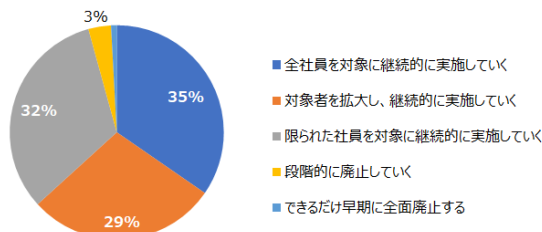


図1 テレワークの継続実施予定

しかし、遠隔コミュニケーションには様々な問題が存在する。通信環境が悪いとうまく繋がらない、疎外感がある、データのやり取りはできても実物がないので触って確かめること、即座に正確に共有し合うことが難しいことなどが挙げられる。遠隔コミュニケーションを発展させるため、これらの問題点を解決する必要がある。

1.1.2 仕事の効率化

遠隔コミュニケーションは、技術職や運用現場で大いに活躍する。多くの企業で、人手不足や人材不足の問題を抱えている。高度な人材が不足しているため、人材育成もうまくいっていない企業は多い。しかし、人材不足のため、熟練者がいない状況での作業や、緊急事態における単独での作業をしなければならない場合がある。そこで、遠隔地から支援することで時間の短縮につながり、効率的に作業を行うことができる。また、ヒューマンエラー防止のため二重確認をする企業も多いが、現場に赴く必要がなくなり、同時に複数の現場の支援を行うことができるので、人手不足の問題点を解決する。

1.2 研究目的

この研究の目的は、離れた場所から作業支援をする際の言葉とカメラだけでは指示を伝えにくい遠隔コミュニケーションの問題点を解決することである。提案するシステムはスマートフォンやタブレット端末を用いて、離れた人が見ているカメラ映像を見ながら自分の手の動きをそこに投影して手を相手に見せることを可能にする。例えば難しい機械操作などをする際に名前を知らない部品があった場合、その部品の特徴を言葉で説明して操作させる代わりに指でさして手の動きを見せるだけで指示をすることができる(図2)。このように、このシステムは遠隔地での作業支援においてより直感的かつ素早く行うことを手助けする。

†和歌山大学大学院 システム工学研究科

‡和歌山大学

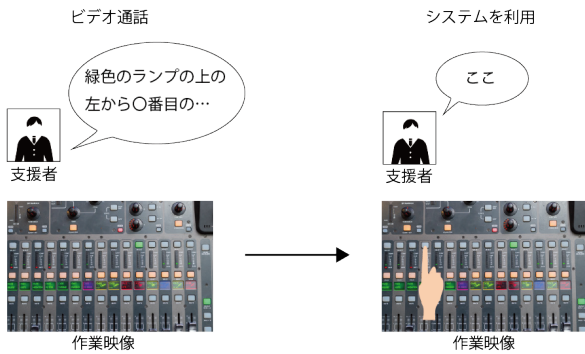


図2 使用イメージ

1.3 関連研究

ARを利用する遠隔支援の研究やシステムは多数存在する。

株式会社日立製作所の「Doctor Cloud」[2]は既存するシステムである。このシステムは指示内容を事前に用意し、マーカーを設置する。作業者はタブレット端末をマーカーにかざすと操作手順を見ることができるだけでなく、作業の対象にカメラをかざすことで現実の物体に指示が重畳される。1度指示内容を作成するとずっと使い続けることができる点は利点である。しかし、作業内容が変わるたびに作り変える必要があり、入念な準備が必要なので緊急でマニュアルとは異なる動きや異なる作業場になると対応できない。また、大きなマーカーを設置する必要があるため、小さい作業場に利用するのは困難である。

加藤晴久らによる研究[3]ではスマートグラスを用いて遠隔支援をする。スマートグラスとは眼鏡型のウェアラブルデバイスで、透過型ディスプレイを用いているので実際の景色に重ねて情報を見ることができる。作業側がリアルタイム映像を送信し、そのキャプチャ画像に支援側が指示内容を書き込んでそれを送信すると、キャプチャ画像の特徴点からマーカーレスで現実物体に指示内容が投影される。スマートグラスが移動しても指示内容は作業対象に重畳し続けることができる。利点は事前準備やマーカーを一切必要とせず、作業に行き詰まった場合にその時々合った指示を受けられる点である。しかし、1度の指示で伝わらなかったり指示が複雑で書かなければならないことが多くあったりすると、その書いて送る作業が手間である。

2. 提案手法

2.1 提案システムの概要

提案するシステムは、スマートフォンやタブレット端末を用いて、一方は手のトラッキング情報を、もう一方はリアルタイム映像を互いに送信し、映像に手を投影したAR

動画の作成と共有を行う。また、それをiOS用のアプリとして開発する。

図3は操作手順である。作業側はスマートフォンやタブレット端末で作業対象を撮影し、リアルタイム映像を支援側へ送信する。支援側もスマートフォンやタブレット端末を用いて、内蔵カメラの前に手を差し出すと送信された作業側の映像に手が上から描画されるので、映像を見ながら手で指をさして操作手順を説明したり、手を動かして動きの見本を見せたりする。その手の動きは作業側へ送信され、作業側の映像に手が描画されることで同じ映像を共有し、その手に従って作業をすることができる。

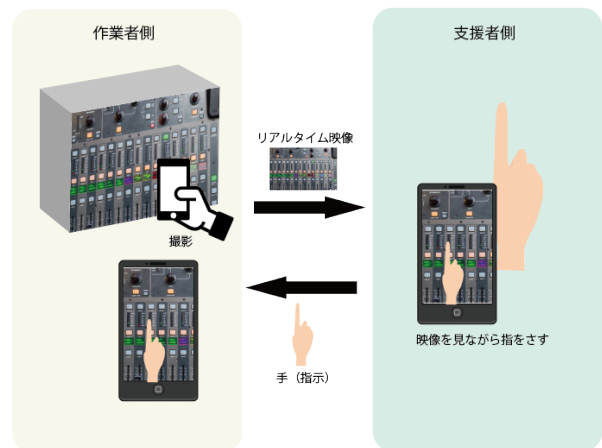


図3 操作手順

2.2 実装手順

作業側と支援側で2つのアプリを作成する。支援側のアプリではハンドトラッキングを行い、UDP通信でもう一つのアプリにトラッキングした手のデータを送信する。また、別のスレッドで映像を受信し表示させる。その受信した映像の上から手を描画する。作業側のアプリでは内蔵カメラで撮影し、リアルタイム映像を表示する。また、UDP通信で手のデータを待ち受ける。受信したデータより手を映像の上から描画する。また、ハンドトラッキングにはMediaPipe[4]を、映像の送信と表示にはSkyWay[5]を利用した。これらはSDKであり、詳細は後述する。システム構成図を図4に示す。

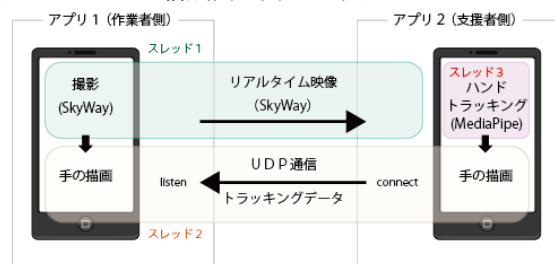


図4 システム構成図

2.3 使用するフレームワーク

2.3.1 MediaPipe

MediaPipe とは Google が提供する機械学習パイプラインを構築するためのフレームワークである。MediaPipe を使用することで、知覚パイプラインをモジュールコンポーネントのグラフとして構築する。iOS や Android, Windows など利用可能なクロスプラットフォームであり、顔検出やフェイスマッシュ、物体検出など 15 のソリューションが提供されている。その中のハンドトラッキングが可能な「Hands」を本研究では利用している (図 5)。

「Hands」は手と指の追跡ソリューションであり、機械学習を利用して1つのフレームで手の 21 個のランドマークを 3 次元座標で推測する。その精度は手のひら検出で平均 97.5%を達成しており、手を握るなどして手のある部分が見えなくても追跡が可能である。また、21 個のランドマークは図 6 のように定義されている。

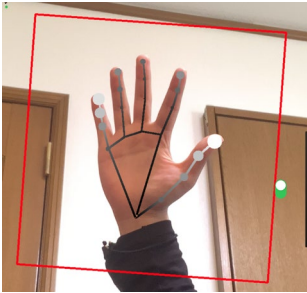


図 5 MediaPipe デモ

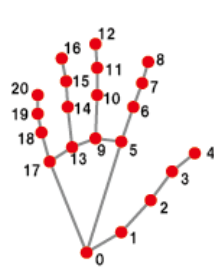


図 6 ランドマーク

2.3.2 SkyWay

SkyWay とは WebRTC をアプリケーションに実装できるクロスプラットフォームな SDK である。JavaScript, iOS, Android の 3 つの SDK が用意されている。WebRTC に必要な 4 つのサーバ (シグナリングサーバ, STUN サーバ, TURN サーバ, SFU サーバ) を API 経由で利用可能である。SkyWay を導入することでテレビ電話やデータ通信などの機能を簡単に自作アプリに実装することができる。本研究では、リアルタイム映像の送信にあたって SkyWay の画面共有機能を利用した。

2.4 開発環境

ハードウェア : MacBook Air (macOS catalina 10.15.4)

ソフトウェア : Xcode 12.4

使用言語

Xcode : Swift

MediaPipe : Objective-C

対象デバイス : iOS12 以上

2.5 実装

2.5.1 映像共有

SkyWay は GitHub でビデオ通話のサンプルプログラムを公開している [6]。実装するにあたって、このサンプルプログラムをベースに他の機能を追加していく。

2.5.2 ハンドトラッキング

MediaPipe は Objective-C で書かれているので、swift と連携させる必要がある。noppefoxwolf が GitHub で MediaPipe の機能の HandTracker のみを抽出し、iOS アプリに組み込めるように Swift と連携させたライブラリ「HandTracker」を公開している [6]。このフレームワークをインポートし、プログラムコードを参考にハンドトラッキングを導入した。

2.5.3 映像共有・アプリケーションの外枠?

トラッキングデータの送信には UDP 通信を用いる。UDP 通信の実装には Network.Framework を使用した。MediaPipe のハンドトラッキングにより得られる 21 個の 3 次元座標をカンマで区切って一つの文字列にし、手のデータ 1 つ分を 1 つのバケットとして送信する (図 7)。支援者側のアプリでは MediaPipe の導入によりあらかじめ備わっているマルチスレッドな関数内で送信する。作業側側のアプリでは SkyWay で映像の共有を始めるボタンを押すことで動く関数の中で待ち受けを開始する。

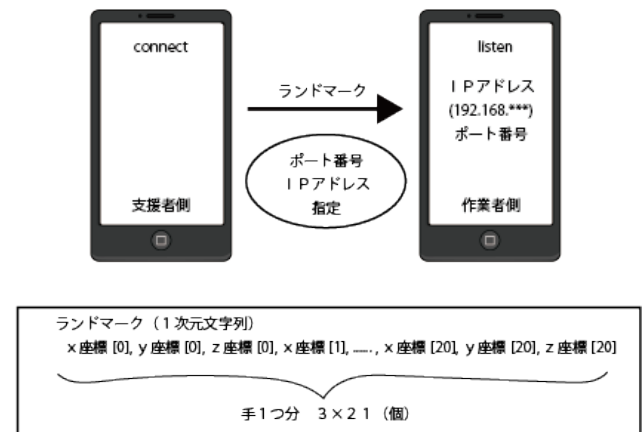


図 7 UDP 通信

作業側側のデータの待ち受けは作業側側がボタンを押すことで開始される。映像の共有は両方のアプリで開始することができるが、手の共有は作業側側が行わなければならない。また、UDP 通信の特徴として、データの送信が一方通行であるためにパケットの紛失やパケットの順番が正しくない場合がある。その場合のデータの修復を行うプログラムを実装していない。これらを改善する必要がある。

2.5.4 手の描画

MediaPipe で得られるランドマークの座標値は以下の範囲で与えられる。

- x 座標 $-1 < x < 1$ (正規化)
- y 座標 $-1 < y < 1$ (正規化)
- z 座標 $-50 < y < 50$ (確認できたおおよその範囲)

SkyWay の画面共有を導入することで作業側側の映像が両方のアプリに表示される。その映像を表示している UIImageView と同じ大きさの View1 をあらかじめ用意し、Storyboard の中で UIImageView の上に重ねて配置する。

View1 の SubView として同じ大きさで新たに生成する View2 に手を描画する。ランドマークが更新されるたびに View1 の SubView(View2)を削除し、もう 1 度 SubView を生成して新たな値で手を描画することで手のアニメーションを実現する (図 8)。

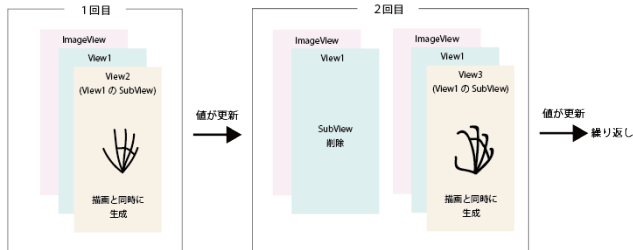


図 8 描画方法

手の描画には UIBezierPath を利用した。手を描画するためのクラスを用意し、そこで MediaPipe の定義に従って点を直線で結ぶ関数を作成した。描画する際の座標値は View1 の高さ と幅を取得し、正規化された値と掛けることで取得する。また、z 座標は数値が大きいほど手は手前にあり、小さいほど手は奥にあることを表す。z 座標を描画する線の RGB 値に反映させることで、手前にあるほど白く、奥にあるほど黒く描画されるように設定した (図 9)。手を描画する関数は、作業側側のアプリではデータが送られると同時に呼び出し、支援側側のアプリでは MediaPipe に備わっている座標を表示する関数内で呼び出す。

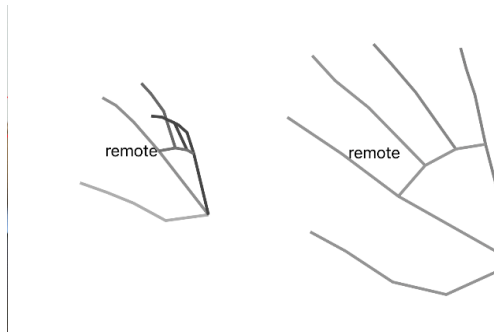


図 9 描画のデモ

2.5.5 UI

SkyWay のビデオ通話のサンプルプログラム [5] をベースに実装を行っているため、UI デザインや画面遷移は変更していない。画面遷移とボタンの挙動は図 10 に示す。作業側側と支援側側の UI は同じである。

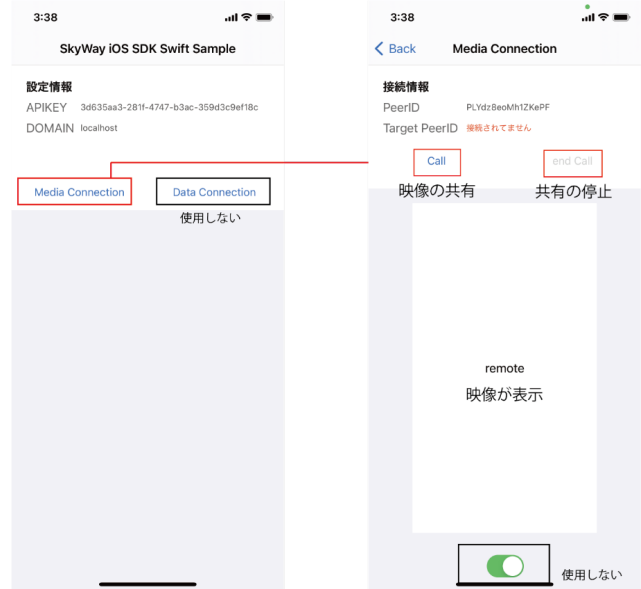


図 10 画面遷移

図 11 は支援側側のアプリを iPad Pro で起動させた場合の画面である。使用する端末により映像を映す UIImageView と手を描画する View の位置がずれてしまうため、画面サイズの変化に対応しなければならない。

また、元から備わっているビデオ通話の機能で本システムとは関係のないボタンが残っているため、無駄が多い。

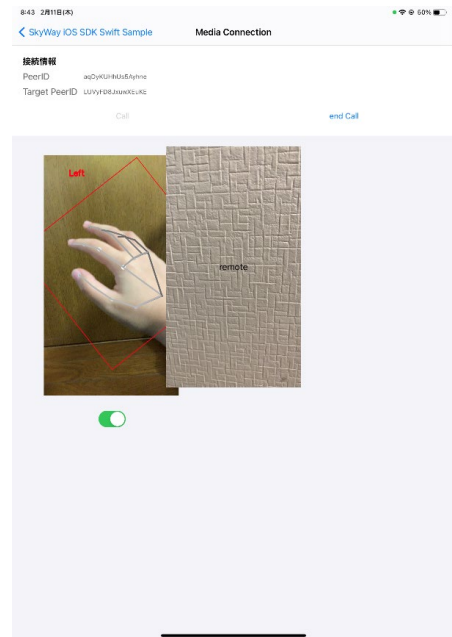


図 11 iPad Pro の場合

2.6 完成したアプリケーション

図 11 と図 12 が実際に完成したシステムを動かした際の画像である。図 11 で撮影している映像が図 12 の支援者側のアプリに写され、円で囲んだ部分のように手が同じように表示される。

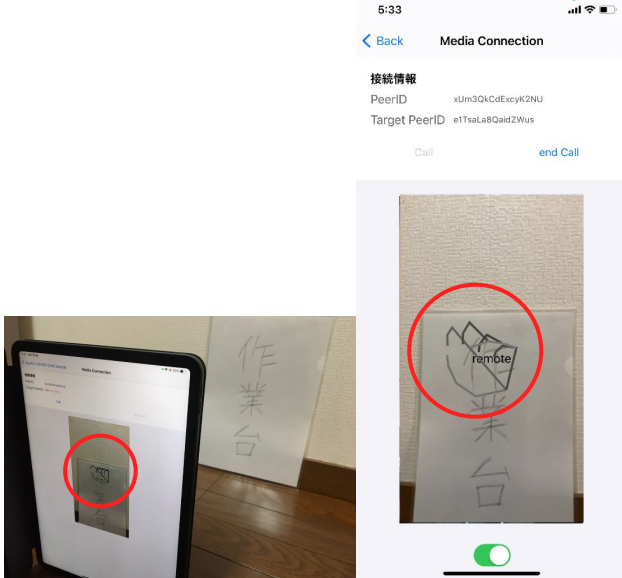


図 12 作業側

図 13 支援側

MediaPipe のデモでは指がぶれることはなかったのに対し、常に指がぶれている状態である。手を描画する際に工夫が必要であると考え、全てのトラッキングデータをその値のまま描画するのではなく、外れ値を除外、または一次ローパスフィルタを用いてデータのぶれを軽減するなどを今後実装予定である。

3. 実験と評価

3.1 実験目的

本システムは遠隔地での作業支援が目的である。そのため、支援者側が意図した動きを作業側が行うかが重要である。この実験で正確性と伝達速度を計測し、本システムを利用しない場合と比較して評価を行う。また、このシステムを用いることで、直感的に指示、または指示を受け取ることができているかを調査する。

3.2 実験方法

図 14 のように円、長方形、正方形の 3 種類の形の色画用紙を各 6 色 2 セットの 36 枚用意する。これらを机に適当に並べ、作業場のボタンであると仮定する。実験では、支援者側が指定した画用紙を作業側が指をさす。連続で 3 回指示を出し、作業側が正しく画用紙を 3 回指をさし終えるまでに間違えた数と経過時間を計測する。その作業を 5 回繰り返し、それぞれの平均値を求め、その数値で評価する。また、指示する画用紙は作業側の意志が入らな

いようあらかじめ乱数で数字を生成し、その数字の画用紙を指示している。

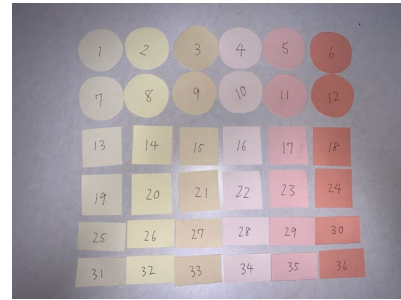


図 14 実験に使用する画用紙

本システムを利用しない場合の指示方法は、ビデオ通話を通し、色や形、場所などの画用紙の特徴を言葉のみで説明する。このとき、数字を答えることは禁止とする。また、画用紙は毎回並べ直して行う。本システムを利用する場合の指示方法は、画用紙の特徴は一切伝えず、指をさすことだけで指示をする。

本システムは指示対象（この実験の場合画用紙）が密集している場合、正確性に欠けると予想したため、画用紙同士の間隔を広げた場合（図 15）と狭めた場合（図 16）の 2 種類の実験を行った。

また、最後に被験者からそれぞれの指示方法に対する意見をヒアリング調査する。

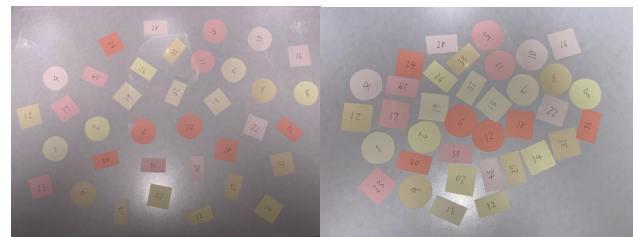


図 15 間隔：広い

図 16 間隔：狭い

3.3 実験結果

この実験を 6 人で行い、2 人 3 組で作業側と支援者側をそれぞれ 1 回ずつ計 6 回計測した。結果を表 1-4 に示す。

表 1 指示方法：言葉 間隔：広い

	1回目	2回目	3回目	4回目	5回目	平均値
経過時間	20.70	21.34	14.24	18.98	19.20	18.89
間違えた数	1	1	0	1	1	0.8
経過時間	18.33	17.26	22.66	8.42	17.72	16.88
間違えた数	0	1	0	0	0	0.2
経過時間	18.29	18.90	12.60	15.01	15.93	16.15
間違えた数	1	0	0	0	0	0.2
経過時間	9.23	20.35	11.53	13.80	19.00	14.78
間違えた数	0	1	0	0	0	0.2
経過時間	13.03	13.65	19.60	12.34	15.56	14.84
間違えた数	0	0	2	3	1	1.2
経過時間	13.43	7.90	14.74	10.36	21.42	13.57
間違えた数	2	0	0	0	3	1.0

表2 指示方法：システム 間隔：広い

	1回目	2回目	3回目	4回目	5回目	平均値
経過時間	15.6	15.2	14.9	15.6	10.78	14.416
間違えた数	0	0	0	0	0	0
経過時間	12.32	17.3	14.17	17.43	14.36	15.116
間違えた数	0	1	0	0	0	0.2
経過時間	12.13	12.27	17.81	11.63	12.32	13.232
間違えた数	0	0	1	0	0	0.2
経過時間	13.57	18.55	17.23	12.18	14.29	15.164
間違えた数	0	0	0	0	0	0
経過時間	16.03	16.82	14.88	15.2	16.58	15.902
間違えた数	0	0	0	0	0	0
経過時間	11.84	15.56	14.28	13.62	22.43	15.546
間違えた数	0	0	0	0	1	0.2

表3 指示方法：言葉 間隔：狭い

	1回目	2回目	3回目	4回目	5回目	平均値
経過時間	10.28	15.66	13.12	13.18	14.1	13.268
間違えた数	0	2	1	1	0	0.8
経過時間	14.62	15.84	11.02	21.42	15.2	15.62
間違えた数	0	0	0	0	0	0
経過時間	28.43	18	23.27	26.05	14.17	21.984
間違えた数	4	1	1	0	0	1.2
経過時間	10.9	10.86	15.3	11.09	13.45	12.32
間違えた数	0	0	1	1	0	0.4
経過時間	13.47	13.82	12.96	12.18	11.77	12.84
間違えた数	1	0	0	0	0	0.2
経過時間	12.53	9.78	11.4	16.35	8.38	11.688
間違えた数	0	0	0	1	0	0.2

表4 指示方法：システム 間隔：狭い

	1回目	2回目	3回目	4回目	5回目	平均値
経過時間	6	12	13.07	13.12	9.3	11.71
間違えた数	0	0	0	0	0	0
経過時間	11.42	14.48	16	12.9	11.2	13.2
間違えた数	0	0	0	0	0	0
経過時間	12.98	12.85	14.63	17.93	11.63	14.004
間違えた数	0	0	0	1	0	0.2
経過時間	10.77	10.57	14.98	19.12	11.55	13.398
間違えた数	0	0	1	1	0	0.4
経過時間	12.52	12.45	8.18	9.8	10.48	10.686
間違えた数	0	0	0	0	0	0
経過時間	12.3	10.35	9.09	9.23	9.4	10.074
間違えた数	0	0	0	0	0	0

画用紙の間隔が広いときと狭いときのそれぞれで、システムを使わずに言葉のみで指示した場合とシステムを利用して指示した場合の結果を比較するためにまとめたのが表5と表6である。システムを利用した場合に数値が低くなった部分を赤で示し、逆に数値が高くなった部分を青で示している。これらの数値は指示が伝達するまでの経過時間と間違えて伝達した数であるため、赤が多いほど良いといえる。

実験結果として、ほとんどの場合において数値が下がり、本システムを利用することで正確率、伝達速度共に向上した。また、間隔は狭くてもほとんど正確に伝達が可能であることがわかった。

表5 比較 間隔：狭い

		言葉	システム	差
1回目	経過時間	18.892	14.416	-4.476
	間違えた数	0.8	0	-0.8
2回目	経過時間	16.878	15.116	-1.762
	間違えた数	0.2	0.2	0
3回目	経過時間	16.146	13.232	-2.914
	間違えた数	0.2	0.2	0
4回目	経過時間	14.782	15.164	0.382
	間違えた数	0.2	0	-0.2
5回目	経過時間	14.836	15.902	1.066
	間違えた数	1.2	0	-1.2
6回目	経過時間	13.57	15.546	1.976
	間違えた数	1	0.2	-0.8
平均値	経過時間	15.852	14.896	-0.956
	間違えた数	0.6	0.1	-0.5

表6 比較 間隔：広い

		言葉	システム	差
1回目	経過時間	13.268	11.71	-1.558
	間違えた数	0.8	0	-0.8
2回目	経過時間	15.62	13.2	-2.42
	間違えた数	0	0	0
3回目	経過時間	21.984	14.004	-7.98
	間違えた数	1.2	0.2	-1
4回目	経過時間	12.32	13.398	1.078
	間違えた数	0.4	0.4	0
5回目	経過時間	12.84	10.686	-2.154
	間違えた数	0.2	0	-0.2
6回目	経過時間	11.688	10.074	-1.614
	間違えた数	0.2	0	-0.2
平均値	経過時間	14.62	12.179	-2.441
	間違えた数	0.467	0.1	-0.367

3.4 評価

実験の結果から本研究の目的の1つである、遠隔作業支援をする際の伝達速度の向上は達成することが出来た。作業対象が狭い場合の正確性も向上し、広い場合と変わらなかったため、作業対象の大きさに制限はないといえる。

結果の特徴として言葉での支援は伝達速度にばらつきがあるのに対し、システムでの支援はあまりばらつきがない。言葉での支援のばらつきは、人により説明の向き不向きがあるからだと考えられるが、システムは誰にでも同じように使用することができるので利点といえる。

本研究のもう1つの目的は直感的に作業支援を行うことである。ここで、ヒアリング調査による被験者の意見を列挙する。

● 言葉での支援

- 「前」は奥か手前かわからない
- 相手と自分で色の言い方が違って難しかった
- カメラに全部映ってないと見落としがあって場所を言い間違えてしまう

● システムでの支援

- 画面の端が指示しにくい
- 確信を持ってない
- 指がぶれて難しい

言葉での支援で被験者が挙げた問題点は説明のしにくさであるため、システムを利用することでこれらの問題点は解決できている。しかし、システムを利用することで新たな問題点が生じた。

画面の端が指示しにくい点は、手がカメラから見切れてしまうことにより手であると判断されないためである。この問題点はハンドトラッキングの問題であり、精度の向上が必要であるが、本研究では行わない。

指示を受け取る際に確信が持てない点は指がぶれてしまうのも原因の1つであると考え。そのため、指のぶれを改善することが重要だと考える。また、これらの問題は、同時に通話することで確認が取れるので、通話機能を実装すると正確性もさらに向上し、使用しやすくなるを考える。

現時点で指示内容に確信が持てなかったり、難しいと感じたりするならば、直感的に作業支援ができていないと言えないが、これらの問題点を解決することで直感的な作業が可能になると考える。

4. 今後の方針

本研究で達成できなかった直感的な作業支援を可能にするための課題解決を行う。被験者アンケートで明らかになった「指がぶれて難しい」、「指示内容に確信が持てない」という課題に対しては、先述したように手のぶれの改善と通話機能を追加する。外れ値を除外し、ハンドトラッキングのデータを一次ローパスフィルタを用いることで整形して解決する。

また、作業には回す、押す、引くなど様々な操作がある。指の骨格だけではそれらを正確に伝えることは難しい。様々な操作を一目で伝達するために、それぞれの動きに対するエフェクトの追加を行う。ハンドトラッキングでは3次元で座標を取得しているため、それぞれの動きを検知して区別することは可能であると考え。手の動きを検知するための初期位置をどのように設定するかは、特定の動作を行ったときなど、深度センサやハンドトラッキングの精度を考慮したうえで今後検討する。

5. おわりに

本研究では、遠隔地での作業支援において、直感的かつ円滑に支援することを目的とし、ハンドトラッキングを利用した新たなシステムを開発した。実験を行った結果、指をさして場所を示す支援において、解決しなければならない問題はあるが、正確性と伝達速度においてともに向上するという今後期待できる良い結果を得ることが出来た。

参考文献

- [1] HR 総研 (2020), “～テレワーク実施企業の 96%が今後も継続の方針、ジョブ型雇用の拡充にも影響か～”, https://www.hrpro.co.jp/research_detail.php?r_no=279 (2021 年 7 月 17 日確認)
- [2] 株式会社日立製作所, “Doctor Cloud/巡回・点検支援システム” http://www.hitachi.co.jp/products/infrastructure/product_site/doctorcloud/ (2021 年 7 月 17 日確認)
- [3] 加藤晴久, 小林達也, 辻智弘, 菅野勝, 柳原宏昌, “スマートグラスの AR 表示による遠隔フィールド作業支援システムの開発”, 映像情報メディア学会誌, Vol.71, No.1, pp. J35-J43, 2017
- [4] “MediaPipe HP” <https://mediapipe.dev/> (2021 年 7 月 17 日確認)
- [5] “SkyWay HP” <https://webrtc.ecl.ntt.com/> (2021 年 7 月 17 日確認)
- [6] “GitHub - skyway/skyway-ios-sdk: iOS SDK for SkyWay” <https://github.com/skyway/skyway-ios-sdk> (2021 年 7 月 17 日確認)
- [7] “GitHub - noppefoxwolf/HandTracker: Detect 3d hand landmark from video stream.” <https://github.com/noppefoxwolf/HandTracker> (2021 年 7 月 17 日確認)