

修士論文

仮想平面を利用した 飛行ロボット操作システムの提案と評価

Proposal and Evaluation of Flying Robot Manipulation System
Using a Virtual Plane

平成 26 年 8 月 14 日 提出

指導教員 小川 剛史 准教授

東京大学大学院
工学系研究科 電気系工学専攻 融合情報学コース
37-126940 米澤 和也

要旨

AR.Droneのような飛行ロボットが容易に入手でき、人々にとって身近な存在となっている。しかし、ロボットを任意の位置へ自由に操作することは容易ではなく、特に飛行ロボットは平面のみを移動するロボットと比較しても操作はより困難である。そこで、飛行物体に重畳表示した仮想平面を用いた飛行物体の操作手法を提案し、プロトタイプの開発を行った。提案手法では、仮想平面上に移動ルートを描くと、そのルートを飛行物体が移動する。ジョイスティックやラジコン操縦無線機のようなコントローラと比較して、レバーやボタンなどを用いた複数の操作を同時に行う必要がなく、飛行物体の向きと操作者の視点を考慮する必要もないため、提案手法では操作の簡易性と直感的な操作性が期待できる。本論文では開発したプロトタイプの操作性が、既存のジョイスティックコントローラと比較して優れているか調査を行った。被験者実験によって、ユーザと飛行ロボットの方向関係によらない、直感的かつ簡易な操作であることが示された。またジョイスティックコントローラでは同時に複数の操作を必要とする、前後左右上下2軸以上の移動を行う際、提案手法の操作は特に優位である結果が得られた。実験を踏まえ、提案手法の特徴を検討し、今後解決すべき課題を考察した。

Abstract

People can get flying robots such as the AR.Drone easily, so robots became familiar with the public. However, it is more difficult for a novice to move positions of flying robots freely than those of robots which move on a plane. In this paper, I propose the flying robot manipulation using a virtual plane by augmented reality. For example, after an operator drew the line on the plane, the robot moves along it. My method enables easier and more intuitive manipulation than methods using controllers such as joysticks or proportional radio control systems, because my method do not involve a complex handling and need not to consider the directions of the robot and the operator. Moreover, I developed the prototype and investigated whether the operativity of it is excellent as compared with a joystick controller. Experiment results showed that the proposed method is not dependent on the directions of the user and the flying robot, and it is the intuitive and simple operation. In addition, it was shown that operation of the proposed method is excellent when the flying robot performs movement of 2 axes or more which needs two or more operations simultaneously by a joystick controller. Based on the experiment results, I considered the features of the proposed method and discussed issues.

目次

第 1 章 序論	1
1.1 研究背景	1
1.2 研究目的	2
1.3 本研究の貢献	2
1.4 本論文の構成	3
第 2 章 関連研究	4
2.1 ユーザの身体と同期させた操作手法	4
2.2 タッチスクリーンを用いた操作手法	6
2.3 移動経路を指示する操作手法	8
2.4 UAV の研究利用	10
2.5 本研究の位置づけ	11
第 3 章 提案手法	12
3.1 提案手法へのアプローチ	12
3.2 操作方法	13
第 4 章 プロトタイプ	16
4.1 ノート PC への実装	16
4.2 タブレット PC への実装	23
第 5 章 評価実験	29
5.1 予備実験：特定の操作タスクを与えた際の所要時間明示と速度制御比較	29
5.2 評価実験：提案手法とジョイスティックコントローラとの操作性比較評価	33
第 6 章 考察	41
6.1 提案手法の特徴	41
6.2 課題	42

第7章 結論	47
謝辞	48
発表文献	49

目次

1.1	飛行ロボットの移動例	2
2.1	TELESAR V[10]	5
2.2	Flying Head[11]	6
2.3	TouchMe[16]	7
2.4	複数台ロボット制御のためのマルチタッチインタフェース [17]	8
2.5	exTouch[19]	9
2.6	Sketch and Run[20]	9
2.7	Laser Gesture Interface[21]	10
3.1	提案手法操作例	13
3.2	提案手法の概略	14
3.3	提案手法における操作	14
4.1	ノート PC 上のプロトタイプ動作	19
4.2	動作フローチャート	20
4.3	方向ベクトルの分解	21
4.4	デッドレコニング	22
4.5	仮想平面の操作方法	25
4.6	着陸を行う高度の表示	26
4.7	自由曲線配列	27
5.1	予備実験：状況設定	31
5.2	AR Flight 操作画面	31
5.3	予備実験：各番号間移動の所要時間	33
5.4	予備実験：移動・旋回操作のみの所要時間	34
5.5	評価実験：状況設定	34

5.6	評価実験：所要時間	36
5.7	評価実験：移動操作のみの所要時間	37
5.8	評価実験：各操作時間の内訳	38
5.9	評価実験：待機時間とインタフェース操作時間の相関	39
5.10	評価実験：操作に十分慣れたユーザの各操作時間の内訳	39
5.11	評価実験：アンケート結果	40
6.1	仮想平面操作の拡張	45

表 目 次

4.1	AR.Drone の仕様	17
4.2	ノート PC 開発環境	18
4.3	タブレット開発環境	24
5.1	評価実験時の操作ミス回数	37

第1章 序論

1.1 研究背景

人が立ち入ることのできない場所の調査や、遠隔コミュニケーションの実現など、テレプレゼンスに関する研究が盛んに行われている [1, 2, 3, 4, 5, 6]. ロボット技術の発展によって、工場内のラインで部品を自動運搬するロボットから一般家庭で利用される Roomba¹ や COCOROBO² などのお掃除ロボットまで、ロボットの活躍の場は着実に拡がり、一般ユーザであってもロボットに接する機会が増加している。

しかし家庭用に普及しているロボットは、移動においてもそのロボット特有のタスクにおいても、あらかじめ決められたプログラムの動作しか行えず、ユーザの望む柔軟な操作を行うことは容易ではない。特に無線で制御できる小型の飛行ロボットは3次元空間を自由に移動できるため、床面に限定された掃除ロボットとは異なり、移動操作に限定しても複数の制御信号を同時にロボットに与えなければならず、その操作は困難である。例えば、ジョイスティックタイプのコントローラを用いてロボットを操作するシステムでは、スティックを倒した方向とロボットの移動方向を対応させている場合が多く、平面上であれば簡単な操作で任意の場所にロボットを移動させることができるが、3次元空間内を移動させる際は平面の移動に加え上下方向の操作が必要となる。一例として、Parrot 社から販売されている飛行ロボット AR.Drone のコントローラは、前後左右移動用のジョイスティックに加えて上昇下降、旋回用のジョイスティックや、離着陸用のボタンで構成されている。また、有人機ではあるが、3次元移動を行う乗り物の代表であるヘリコプターは、前後左右移動用スティック、上昇下降用レバー、旋回用ペダルによって主に操作される。一般的に空間を移動する飛行ロボットは、前後、上下、旋回の3自由度、もしくは左右移動を加えた4自由度の操作がコントローラに割り振られていて、平面移動に比べ複雑な操作となる。また、機体の外部から操作する場合、ロボットにとっての方向と操作しているユーザの方向は一致しないことが多い。そのためユーザの視点で操作を行う場合は飛行ロボットの方向を考慮して行う必要があり、直感的な操作を行うことはできない。したがって、空間の移動操

¹<http://www.irobot.com/us>

²<http://www.sharp.co.jp/cocorobo/>

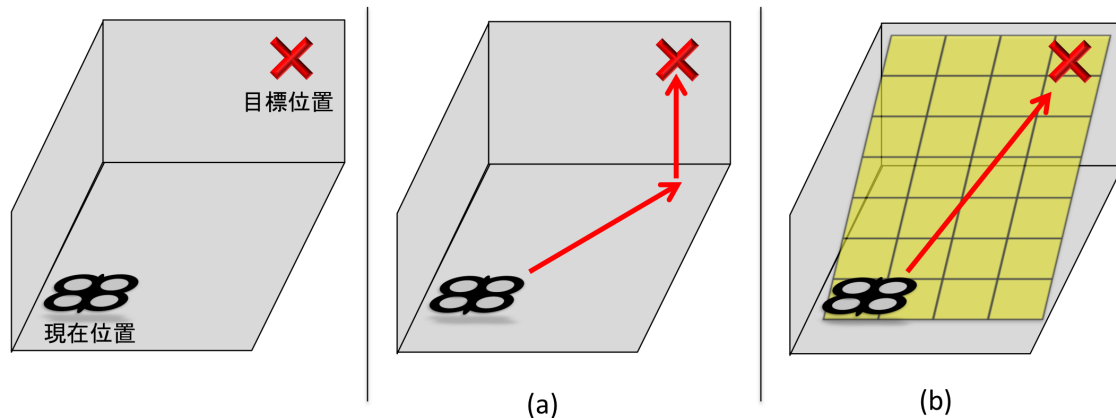


図 1.1 飛行ロボットの移動例

作は困難であり，習熟するには相応の訓練が必要となる。

1.2 研究目的

本研究では，3次元空間を移動する飛行ロボットに対して，直感的で自由に操作可能なインタフェースの実現を目指すために，飛行ロボットの移動ルートを，仮想平面を用いて視覚的に提示する操作手法を提案する．飛行ロボットを中心とした仮想平面上に飛行経路を指示することで，3次元の移動指示を2次元的な操作で行うことが可能となる．仮想平面の表示は拡張現実感 (AR: Augmented Reality) 技術を利用することで行い，操作インタフェースとしてタブレット PC を用いる．移動例として，図 1.1 の様に空間の対角線の位置に移動する場合を考える．既存のジョイスティックタイプのコントローラでは，操作に習熟していない場合，複数の移動操作を同時に行うことが困難であり，コントローラの操作配置によっては図 1.1(a) に示すように平面移動と上昇それぞれの操作を独立して行う必要がある．一方，提案手法ではタブレットのスクリーンを通して，図 1.1(b) に示すように飛行ロボットを移動させたい箇所に仮想平面を調節し，任意の移動経路をタッチ操作で描くことで直線上に目標位置へ向かい，効率的な移動が実現できる．また，これらの操作はユーザと機体の向きに関係なく行えるため，位置関係を考慮することなく直感的に操作できる。

1.3 本研究の貢献

本研究で提案した拡張現実感を利用した飛行ロボット操作手法によって，既存のジョイスティックタイプのコントローラと比較してユーザと飛行ロボットの方向関係によらない，

直感的かつ簡易な操作インタフェースを実現した。またジョイスティックコントローラでは同時に複数の操作を必要とする、前後左右上下2軸以上の移動を行う際、提案手法の操作は特に優位であることを示した。

1.4 本論文の構成

本論文の構成は以下の通りである。第2章ではロボットの遠隔操作インタフェースに関する関連研究を、ユーザの身体と同期させた手法、タッチスクリーンを用いた手法、移動経路を直感的に指示する手法それぞれについて述べ、それぞれの操作手法の課題を議論する。また、本研究で利用する無人飛行機 (UAV: Unmanned Aerial Vehicle) の研究利用について述べる。第3章ではARを利用した飛行ロボット操作手法を提案し、第4章でノートPCとタブレットPCそれぞれのプラットフォームに開発したプロトタイプについて述べる。第5章ではタブレットPC上で開発したプロトタイプを用いて、特定のタスクを与えた際にかかる操作時間の明確化と、既存のジョイスティックタイプのコントローラと操作性の比較を行い、その実験方法と結果を述べる。第6章では実験を踏まえて、提案手法の特徴と課題について考察を行う。最後に第7章で結論を述べる。

第2章 関連研究

本章では，2次元平面，3次元空間の移動ロボットに限らず，開閉アームが搭載された作業ロボットなどの，遠隔操作インタフェースに関する事例について，ユーザの身体とロボットの動作を同期させた操作手法，タッチスクリーンを用いた操作手法，移動経路を指示する操作手法をそれぞれ述べる．また，飛行ロボットとして本研究で想定している UAV について言及し，既存の研究に関する課題について考察する．

2.1 ユーザの身体と同期させた操作手法

3次元空間を移動するロボットの操作手法として，ユーザの視線の動き [7] や脳波を利用した操作 [8]、身体全体のジェスチャに合わせて操作を行う手法 [9] など，ユーザの挙動から直接操作する手法が研究されている．しかし，視線や脳波の生体情報は安定した制御が困難であり，正確な操作は容易ではない．また既存のコントローラの動作をジェスチャにマッピングした操作は，ロボットの移動量を推測することが困難である，そこで，本節では3次元の移動を行う人間の身体動作にロボットの動作を同期させて，直感的に操作する手法について述べる．

2.1.1 TELESAR V

ユーザの上半身の動きとロボットを同期させたシステムとして，TELESAR V[10]がある．遠隔地にいながら，目の前の物に直接接触しているかのような感覚を提供するテレイグジスタンスシステムであり，ロボットの指先が受けた，押下力やせん断力，温度といった刺激をユーザにフィードバックすることが可能である．TELESAR の操作手法はユーザが制御・操作を司り，ロボットがその制御下で動作を行うマスタースレーブ方式であり，ユーザにの上半身に取り付けられたセンサにより，首の向きや，手の動き，指の開閉などの身体動作をキャプチャする．得られた情報は遠隔地に送られ，胴体に 5 DOF，頭部に 3 DOF，腕に 7 DOF，手に 15 DOF，他の箇所を合わせ計 52 の自由度をもったヒューマノイドロ



図 2.1 TELESAR V[10]

ボットはユーザの身体と同期した動きを行う。またロボットに搭載されたカメラの映像をユーザに装着したHMDに映しており、遠隔地の視界を提供することができる。TELESARシステムでは、多くの自由度を活用し、将棋や積み木といった細かいタスクを行うことが可能である。しかし、現状では上半身のみを同期させることしか行えず、下半身同期による移動等を行うことはできない。

2.1.2 Flying Head

Flying Head[11]はUAVの遠隔操作において、操作者の頭部の動きとUAVの動きを同期させる操作方法を提案している(図2.2)。一般に、飛行ロボットを制御するためのパラメータは、高度や前後左右上下方向それぞれのスピード、機体のピッチ角、ロール角、ヨー角など、平面移動のみを行うロボットと比較してとても多い。それらを制御するために、手持ち可能な操作端末(PDA: Personal Digital Assistant)をはじめ、音声認識や模型コントローラなど、様々な操作インターフェースが考案され、評価されてきた[12]。日常的に行われる身体操作を複雑な制御を必要とするUAVの移動にマッピングすることで、UAVの動作とユーザの運動感覚をリンクさせ、直感的な操作が可能となる。ユーザの動作は周囲に設置したモーションキャプチャを行うカメラから得られ、またUAVに搭載されているカメラの映像を、ユーザに装着したHMDに送ることで視点の同期も行っている。Flying Headの特徴として、ユーザは頭部の位置と向きを変化させることにより、UAVにそれらの操作を同時に行わせることができる。しかしUAVの移動範囲はユーザ周囲の環境に依存するため、広い空間での操作を行う際は操作範囲と同様のスペースを用意する必要がある。ま

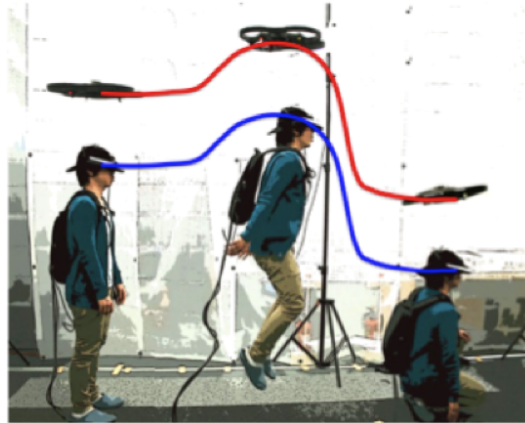


図 2.2 Flying Head[11]

た、垂直方向に関しては身体動作と完全に同期させるため、操作者の身長を超えるような高さに UAV を誘導することは困難である。

2.2 タッチスクリーンを用いた操作手法

ロボットの操作インタフェースとしてスクリーンを直接タッチして操作を行う手法が提案されている [13, 14, 15]。主に、操作対象となるロボットをカメラで捉え、スクリーンに投影されたロボットにタッチ等の操作を加えることで移動操作を行う。スクリーンには操作対象となるロボットと周囲の状況が映し出されるため、移動を阻害する障害物を把握しながら操作を行うことができる。しかし、3次元以上の操作を必要とするロボットを2次元操作インタフェースであるタッチスクリーンを用いて操作することは容易ではない。

2.2.1 TouchMe

タッチスクリーンを用いたロボット操作インタフェースとして、TouchMe[16]が挙げられる。本システムではアームを付けた平面移動ロボットの操作を想定しており、操作対象を含めた作業空間を3人称視点で捉えたカメラ映像がタッチスクリーン上で表示され、ユーザは遠隔地で映し出されたロボットを操作することができる。ロボットの移動操作やアーム操作は3次元的な操作になるが、タッチスクリーンでは2次元的な操作しか行えない。そこで、スクリーンに映し出されたロボット上に操作を補助するためのCG (Computer Graphics)を重畳表示することで、直感的な操作を実現している(図 2.3)。ユーザは画面に表示されたロボットに対し、操作したい位置をスクリーン上でタッチしドラッグすることで、任意

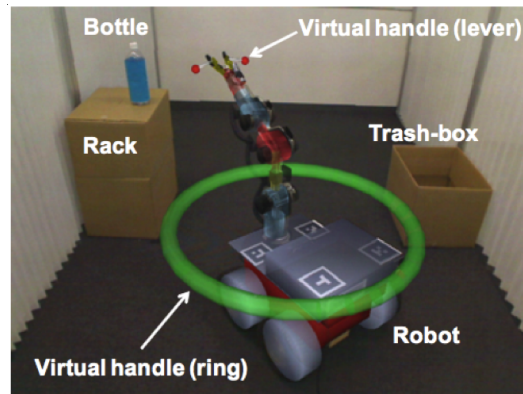


図 2.3 TouchMe[16]

の位置の操作を行うことができる。画像内のロボットには同じ外見のCGが重畳表示され、CGはユーザの入力に応じて動く。こうしてユーザにロボットの動作を予測させることができる。ロボットの前進後退操作は重畳されたCGをロボットの前後方向にスライドすることにより実現し、旋回操作はロボットの周囲にリング状のCGを表示させ、任意の方向にリングをスライドすることで実現する。平面の移動は上記の前進後退、旋回操作を繰り返すことで行う。またロボットに備えられたアームの移動や開閉も旋回と同様にして補助CGを用いて操作する。しかし上昇下降を加えた3次元的な移動操作を行うためには旋回の軸をもう1つ増やし、2軸の旋回操作をタッチスクリーンで行う必要があり、煩雑かつ時間のかかる操作となってしまう。

2.2.2 複数台のロボット操作を行うためのマルチタッチインタフェース

複数のロボットに対して移動経路を指示し操作するマルチタッチインタフェース [17] について述べる。提案システムでは天井に設置したカメラの映像をマルチタッチディスプレイに投影し、映し出された複数台の掃除ロボットに対して、タッチ操作でベクトル場を与えることで、生じたベクトルの向きへ移動を同時に指示することができる(図2.4)。それぞれのロボットに対してARToolKit[18]によるマーカ認識を行うことでそれぞれの位置を取得し、ディスプレイ上のロボットに独立して移動命令を送ることができる。また移動操作のみに限らず、タッチディスプレイに対してピンチ動作を行うことにより掃除ロボットのON/OFFを切り替えることができる。しかし、移動操作は床面に限定され、本システムでは空間を移動するロボットに適用することは困難である。

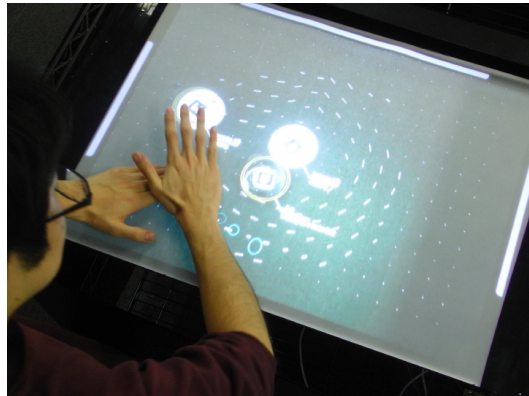


図 2.4 複数台ロボット制御のためのマルチタッチインタフェース [17]

2.2.3 exTouch

exTouch[19] は操作したい対象をタブレット PC で捉えることにより、仮想的な平面をスクリーン上に提示し、ユーザが仮想平面を動かすことで実物体を操作する手法を提案している (図 2.5)。移動させる実物体の一部をマーカとして認識し、それによって得られた仮想平面に対してスライド、回転等の操作を行うことで実物体に対しても移動、旋回等を指示できる。また、仮想平面をスクリーン上でタッチしたまま実際にユーザが移動することでも、実物体は移動した仮想平面を追従し操作が可能となる。それぞれの操作モードにおいて原理は同様で、仮想平面とマーカの現在地の差分をフィードバックして移動を行う。提案システムのプロトタイプとして、360 度全方向に自由に移動できる平面移動ロボットや可動式の家具、そして飛行ロボットを操作することができる。しかし仮想平面の移動と向きの変更を同時に行うことが不可能なため、移動中に平面外の動きを加えることは不可能である。

2.3 移動経路を指示する操作手法

ロボットの移動経路を特定の方法を用いて指示し、その指定経路をロボットが追従することで操作を行う手法は直感的であり、操作が容易である。

2.3.1 Sketch and Run

Sketch and Run[20] は天井に設置されたカメラで床面にある掃除ロボットを捉え、その映像をタブレット PC に投影し、図 2.6 に示すように画面上で移動経路をスタイラスペンを



図 2.5 exTouch[19]

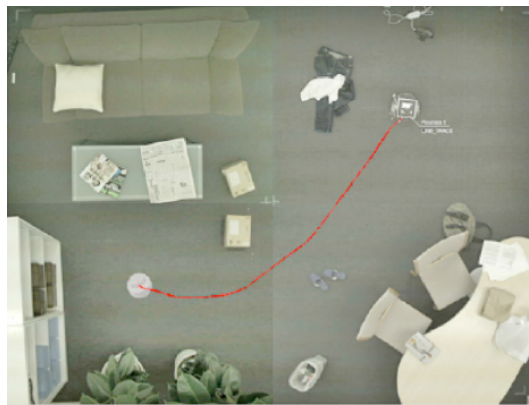


図 2.6 Sketch and Run[20]

用いて記述することにより操作を行う。記述した軌跡は自由曲線で描くことができ、旋回と前進を繰り返しながら記述通りの経路を移動することができる。スタイラスペンでループ状の記述を行うことにより、閉じたループ内の区間を掃除する操作や、特定の記述パターンを描くことで、動作の一時停止、再開、特定の位置に戻る等の操作を行うことを想定している。しかし、操作インタフェースの2次元平面と実空間の床を対応させていることから、設置したカメラの向きと床面は垂直関係を保つ必要があり、移動カメラを用いることは難しく、操作範囲が限定されてしまう。また、カメラの奥行き方向に移動経路を指定できないため、3次元空間の移動操作は不可能である。



図 2.7 Laser Gesture Interface[21]

2.3.2 Laser Gesture Interface

Laser Gesture Interface[21]では、レーザポインタで床平面上に記述した軌跡を用いて、ロボットに対する操作コマンドを指定する方式を提案している。レーザポインタを用い、動かしたい対象を囲むことで操作対象の選択を行い、レーザポインタで床面に自由曲線を描くことで選択した対象物を、ルート通りにロボットが移動させる(図2.7)。ロボットやレーザポインタの軌跡などは、天井に取り付けたカメラで検出している。ユーザが指示したコマンドは、天井のプロジェクタから床面に投影され、その内容を確認できるようになっている。タブレットなどの操作インタフェースを必要としないため、より直接的な操作感覚が得られるが、レーザ光によるコマンドは床や壁などの平面にしか記述できないため、空中での移動といった操作は行えない。

2.4 UAVの研究利用

無人操作ロボットの利用は、ロボットに搭載したカメラや各種センサ等を用いて極高低温、高所等、人間の立ち入りが困難な場所における調査や救助[22]に使われることが見込まれ、特にUAVは床面を移動するロボットに比べ調査範囲が格段に広がり、盛んに研究されている[23, 24]。UAVは固定翼、回転翼の機体の2種があり、後者は空中の一点にホバリングを行うことが可能である。既存の操作手法は一般的に無線により行われ、機体を目視しながらのコントローラ制御、遠隔地からの衛星回線を利用したGPS制御など多様である。UAVは20センチメートルから大きいもので30メートル程度と大小様々であり、人が搭乗していないことから、人が立ち入ることの難しい場所や、単調な作業など多くの利用状況に応じて研究されてきた。UAVに搭載したセンサやカメラを用い、探索を目的とした研究[25]や、農薬散布への利用[26]、災害時の情報収集[27]、空中架線工事の支援、高所からの写真撮影[28]など応用範囲は広い。

2.5 本研究の位置づけ

既存研究では、身体に同期させる手法の場合、ユーザの周囲環境に動作が依存してしまうため、操作範囲に制限が設けられてしまう。またタッチスクリーンを利用した手法や移動経路の指示は簡易な操作であるが、3次元空間への適用が難しい。そこで本研究では飛行ロボットに対して、直感的で自由に操作可能なインタフェースを実現するため、拡張現実感を利用することで3次元空間中に移動経路を記述する、新しい操作手法を提案する。次章では提案手法について述べる。

第3章 提案手法

本章では、飛行ロボットを操作するための要求条件を明らかにし、それらを解決する操作手法を提案する。そして提案手法におけるインタフェースの操作方法を述べる。

3.1 提案手法へのアプローチ

図 3.1(a-1) に示したように、3次元空間の対角線の位置にロボットを移動させたい場合、既存のコントローラでは直線的に目的地に向かわせることは容易ではなく、一例として床と平行な平面移動をした後に上昇を行う、といった複数の行程が必要になってしまう。また飛行ロボットを3人称視点で操作する場合、ロボットの周囲環境が把握しやすい反面、移動方向の指示はロボット主観で行われるため、例えばロボットの前面とユーザが対面している状態で操作を行う場合、ユーザが望む移動方向と実際のロボットの移動方向は前後左右逆に動作してしまう。このように、ロボットとユーザの位置関係によっては、ユーザの想定した動作を行わせることが困難であり、移動により多くの時間を要してしまう。また、2章では、ユーザ動作と同期させる手法は操作範囲が限定されてしまうこと、タッチスクリーンインタフェースを用いる手法は3次元的な操作が困難であること、そしてロボットの移動経路を指定する手法は、描いた軌跡を投影する床が空中に用意できないため、飛行ロボットに適用できないことをそれぞれ問題点として挙げた。さらにロボットをカメラで捉えて操作する手法において、図 3.1(b-1) のように柱の間を通過してロボットを移動させたい場合、障害物に飛行ロボットが阻まれてしまい、操作を続けることは不可能になってしまう。

以上を踏まえて、3次元空間を移動する飛行ロボットに対して、直感的で自由に操作可能なインタフェースを実現するために、移動空間中に仮想的な平面を提示し、その仮想平面上に経路を描くことで飛行ロボットを操作する手法を提案する。提案手法の概略図を図 3.2 に示す。タブレットに搭載されているカメラで飛行ロボットを捉え、ディスプレイに表示された飛行ロボット上に仮想的な補助平面を重畳表示する。仮想平面上にロボットの移動経路を指定し、その経路に従ってロボットが移動していく。

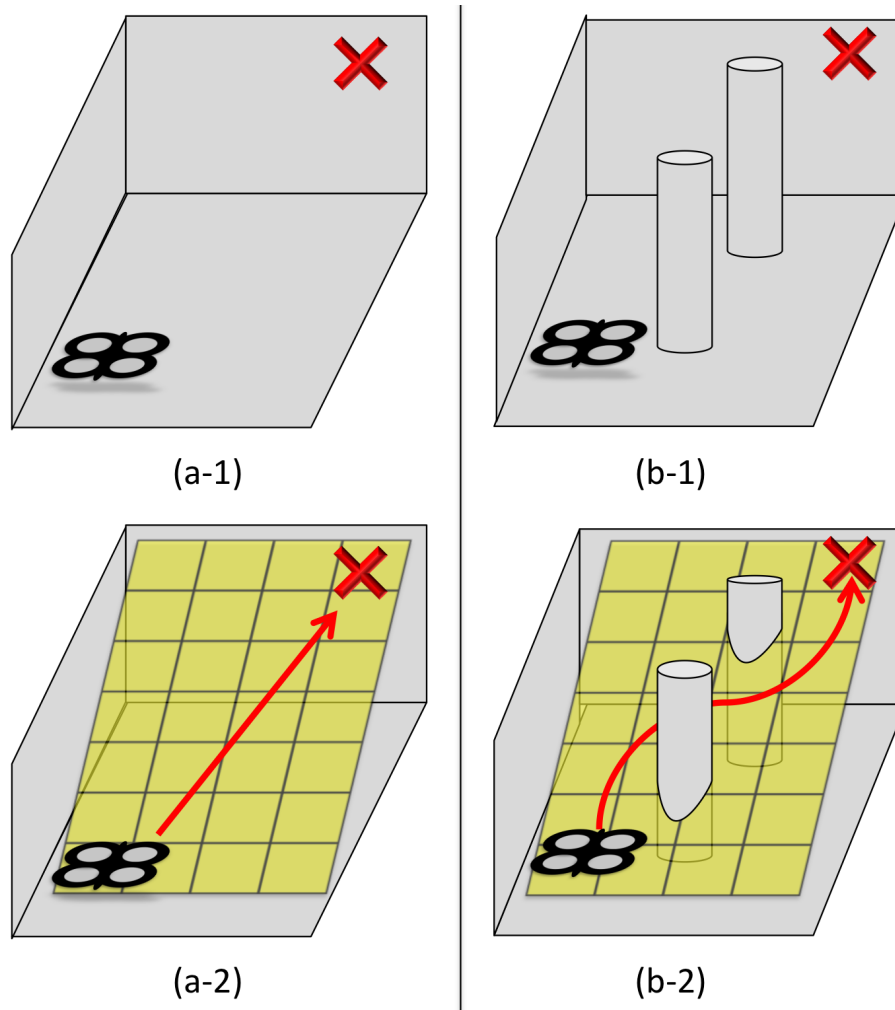


図 3.1 提案手法操作例

図 3.1(a-2) に示すように、提案手法では仮想平面の向きや大きさを目的地を通るように調節し、移動軌跡を指定することで、ロボットとのユーザの方向関係を気にせず、少ない行程で移動を行うことができる。また、一旦移動軌跡の記述を完了してしまえば、図 3.1(b-2) に示すように一時的にロボットが障害物等で視認できなくなっても移動を完遂することができる。

3.2 操作方法

飛行ロボットの基本動作は、前後左右上下方向の移動、離着陸、旋回、動作の中断である。提案手法ではタッチスクリーンを用いた操作指示を想定しており、それぞれの基本動

作に対応するタッチスクリーン上での操作を考案した。操作概要を図 3.3 に記す。以下にそれぞれの操作の詳細を述べる。

移動

ロボットが飛行状態のときに、仮想平面上にロボットの移動ルートを自由曲線で記述する。仮想平面は飛行ロボットを中心に回転でき、タッチスクリーン上のピンチイン/アウト操作で大きさを変更することで、広い範囲での移動指示も可能である。仮

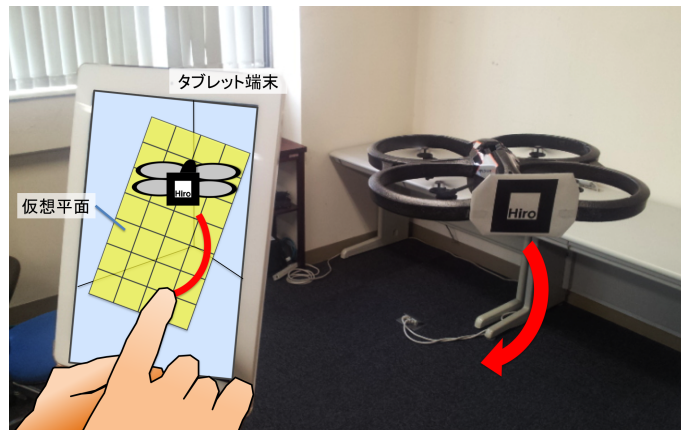


図 3.2 提案手法の概略

	移動	離陸	着陸	旋回	動作の中断
ロボット動作					
画面操作					

図 3.3 提案手法における操作

想平面を用いた移動ルート指定では、ロボットが障害物により一時的に視認できなくなるような状況でも、移動ルートの指示通りに移動を行う。また、飛行ロボットの移動速度は、操作者による移動ルートの記述スピードに対応させ、速く描いたルートは速いスピードで、遅く描いたルートは遅いスピードで移動させる。

離着陸

着地状態の飛行ロボットに重畳表示した仮想平面上へ上方向のルートを記述することで離陸を指示する。またロボットが飛行状態のときに床面との距離が閾値以下になるまで下方向のルートを記述することにより着陸を指示する。閾値の具体的な値に関しては、操作するロボットの仕様にも影響を受けるため任意に指定できるものとし、仮想平面上に閾値の境界を表示する。

旋回

ロボットが飛行状態のときに、2本指で水平方向にドラッグすることで、旋回を指示する。

動作の中断

飛行ロボットが移動している最中に、タッチスクリーンをダブルタップすることで移動動作を中断し、その場でのホバリングを指示する。

第4章 プロトタイプ

本章では提案手法を元に、ノート PC とタブレット PC に実装したそれぞれのプロトタイプについて述べる。また、両プロトタイプに用いる飛行ロボット AR.Drone の仕様を説明する。

4.1 ノート PC への実装

提案手法を元にプロトタイプを作成した。初めはノート PC にシステムを実装し、飛行ロボットに想定動作を行わせることが可能か、また、どのような操作性の特徴があるか検証を行った。完成したプロトタイプの操作プロセスは、まず飛行ロボットを Web カメラで捉え、映像をノート PC に映す。捉えた AR のマーカ上に仮想平面を表示させ、平面の大きさや向きをマウスで操作し、同様にマウスを用いて移動させたい目標位置をクリックする。その位置へ直線的に移動を行う。本節では操作対象に用いる AR.Drone の概要説明、ノート PC に実装したプロトタイプの開発環境、操作インタフェース、AR.Drone の制御方法について述べる。

4.1.1 AR.Drone

本研究で使用している飛行ロボット、AR.Drone について述べる。AR.Drone はタブレットやスマートフォン端末をコントローラとし操作を行うことができる Wi-Fi コントロールヘリコプターである。仕様を表 4.1¹ に記す。表に記載されている通り、AR.Drone にはカメラをはじめ多くのセンサを搭載しており、また開発環境として API もととのっているため、研究開発に多く利用されている [29, 30, 31, 32, 33, 34, 35, 36]。AR.Drone はピッチ角、ロール角、ヨー角、高度、速度等のパラメータを航行データとして逐次取得することができ、また、移動や旋回命令を逐次送信することでソフトウェアから AR.Drone を制御することが可能である。

¹<http://ardrone.tv/ardrone20.html>

重量	455 g
最大飛行速度	5 m/s
飛行時間	約 12 分
内蔵カメラ	HD カメラ 720p 30fps QVGA カメラ 60fps (水平方向の速度測定用)
OS	Linux 2.6.32
搭載コンピュータ	CPU：1GHz32ビット ARM Cortex A8 プロセッサ メモリ：1Gbit DDR2 RAM (200MHz) Wi-Fi b/g/n、2.0USB インターフェース×1
フライトセンサー	3軸の磁気センサー 3軸ジャイロスコープ 対地面の高度測定用超音波センサー
動力	ブラシレスモーター (35,000 回転/分 出力 15 w)
電源	リチウムポリマーバッテリー 2.0 専用 (11.1V 1000mAh)
充電時間	約 90 分

表 4.1 AR.Drone の仕様

4.1.2 通信概要

ノート PC やタブレット PC などの操作端末は、AR.Drone に対して、Wi-Fi 経由の UDP ソケット通信によって、操作命令の送信と航行データの受信を行う。操作命令は AT コマンドと呼ばれる特定の文字列データを送信することで実現する。離着陸、前後左右上下方向への速度指定、旋回方向への回転速度指定、停止等の動作にそれぞれ対応する AT コマンドがあり、インタフェースの操作に応じて逐次送信するコマンドを切り替えている。また AR.Drone は 50 ミリ秒以上信号を受信しなかった場合は自動的にその場で移動を停止し、ホバリングを続け、2000 ミリ秒以上信号を受信しなかった場合は通信が終了してしまうため、プログラム上では 50 ミリ秒未満の間隔でコマンドを送信し続ける必要がある。

AR.Drone にはジャイロセンサ等多くのセンサが搭載されており、速度や高度、姿勢角度等の航行データとなるバイナリデータを取得することが可能である。本システムでは操作命令を送る時とは別のポートを用いて UDP 通信を行い、30 ミリ秒間隔で送信命令と同様に航行データを受信している。得られたデータは 4.1.6 節で述べるデッドレコニングや、4.2.2 節で述べる操作インタフェースに利用している。

ノート PC	MacBookAir OSX 10.8.5 (Apple 社)
プログラミング言語	Processing 1.5
AR 環境	ARToolKit
Web カメラ	HD Pro Webcam C920t

表 4.2 ノート PC 開発環境

4.1.3 開発環境

ノート PC に実装した開発環境を表 4.2 に記す。Processing のライブラリとして、マーカ認識と座標変換を行う ARToolKit とレンダリングするオブジェクトを決定する OpenGL, AR.Drone を制御するライブラリ²を使用している。

4.1.4 操作インタフェース

AR.Drone の前後左右 4 面に画像マーカを搭載し、それらを Web カメラで検出して、PC のディスプレイに黄色い半透明な仮想平面を重畳する。仮想平面は、画像マーカの原点を通る平面となっており、マウスの右ボタンを押したままドラッグすることで回転操作を行う。また、初期状態では、仮想平面はマーカと同じ大きさで表示されるが、AR.Drone を移動させたい目標位置まで平面の大きさが届いていない場合が考えられるので、マウスのホイールを用いることでユーザが任意に平面の拡大縮小を行えるようにしている。仮想平面の向き、大きさを決定した後、平面上の移動させたい位置を左クリックすることで現在位置から目標位置まで赤色の直線が表示され、その直線上を飛行ロボットが移動する。飛行ロボットの移動は全て仮想平面上で行われることを想定しており、仮想平面の向きを任意に変更することで 3 次元空間の移動が可能となる。また、移動を開始した時点の仮想平面の位置はスクリーン上で固定されるように設定し、その際に平面の透明度を上げているため、AR.Drone の移動に仮想平面は伴わず、指定した経路上を移動している様子が視覚的に分かるようになっている。目標位置に到達した時点で仮想平面は再度マーカ上に表示される。飛行ロボットが移動する様子を図 4.1 に示す。また、本プロトタイプでは離着陸はキーボードのボタン操作により行い、提案手法で想定していた旋回、動作の中断操作は実装していない。

²<http://kougaku-navi.net/ARDroneForP5/>

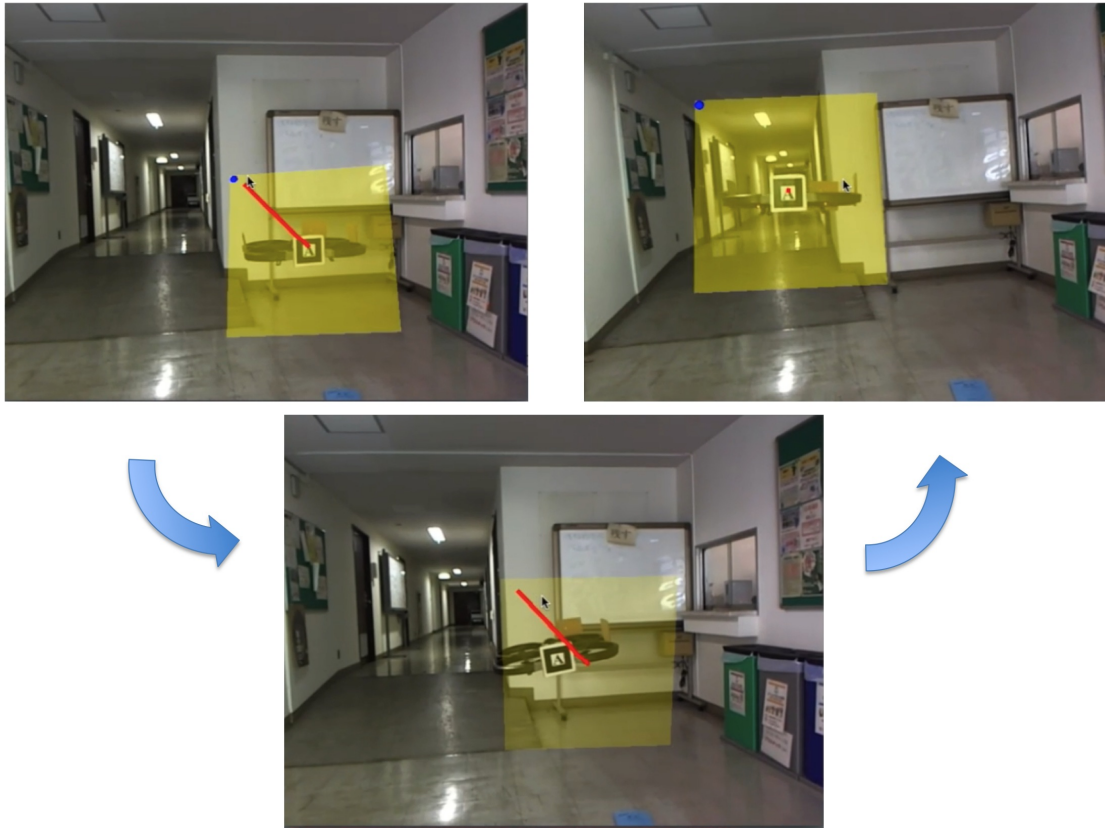


図 4.1 ノート PC 上のプロトタイプ動作

4.1.5 AR.Drone の制御

本節では現在位置からクリックした目標位置への移動に際して、どのようなプロセスで送信する命令を決定しているか述べる。移動制御における処理の流れを図 4.2 に記す。Stay モードと Move モードの 2 種類の状態を定義し、Stay モードでは、現在位置に停止（ホバリング）するコマンドを、Move モードでは特定の方向に移動するようコマンドを送信する。移動指示は、仮想平面上の移動させたい位置をクリックした時点で Move モードに切り替わり、目標方向への移動コマンドを与えることで実現する。一連のプロセスでは、3次元のカメラ座標系における AR.Drone の現在位置の座標と、仮想平面上に記した目標位置の座標を利用して、現在位置はマーカの中心点から、目標位置はタッチ位置と仮想平面の交差点から、それぞれ算出している。カメラ座標系の 2 点の座標を、移動直前の AR.Drone の位置を原点とした空間座標系に変換し、2 点間の距離を近づける方向に移動を行う。Move モード中、現在位置は後述するデッドレコニングで逐次算出する。

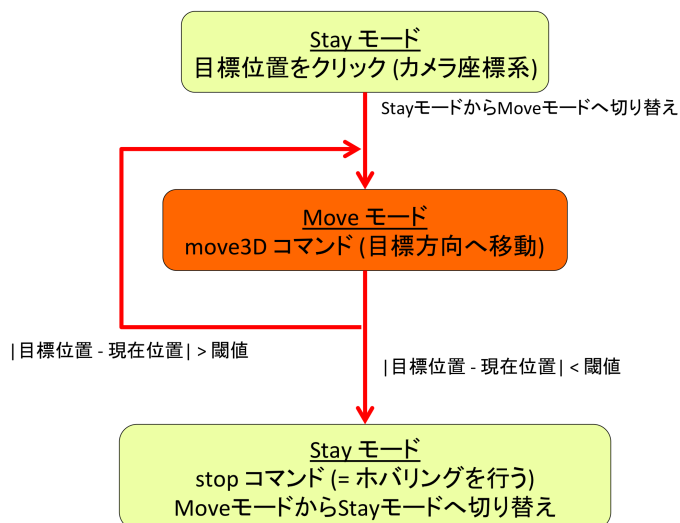


図 4.2 動作フローチャート

移動を開始した後、AR.Drone が目標位置に到達するまで方向を修正しながらコマンドを送信し続け、現在位置と目標位置の座標値の距離差が閾値以下となったところで、AR.Drone に stop コマンドを送り、Stay モードへ切り替える。上記の判定は 1/60 秒毎に行っている。また AR.Drone の移動速度は 85 センチメートル毎秒としているため、理論的には約 1.4 センチメートル移動する毎に修正された方向への移動指示やホバリングといったコマンドが送信される。しかし実際には、AR.Drone が stop コマンドを受信してからホバリング状態に移行するまでに慣性により 10 センチメートル程度移動してしまう。そこで Move モードから Stay モードに切り替える座標の閾値を予備実験で求め、目標位置から 20 センチメートル以内、すなわち目標位置を中心とした直径約 40 センチメートルの球状内に留まるよう設定した。また、安全のため一定時間移動した後自動的に Stay モードになり停止するよう処理している。

方向指示制御

4.1.2 節で記した通り、AR.Drone は各 3 軸方向の速度を指定することで任意方向の移動を実現する。各軸それぞれの速度を決定するために、まずは図 4.3 に示すように現在位置と目標位置の座標差となる方向ベクトル \vec{d} を求める。求めたベクトルを AR.Drone の各方向成分に分解し、それぞれの方向速度を以下の式で決定する。この時、速度 V は定数であり、常に一定速度で移動を行う。また AR.Drone の仕様により、上下方向の速度のみ他方

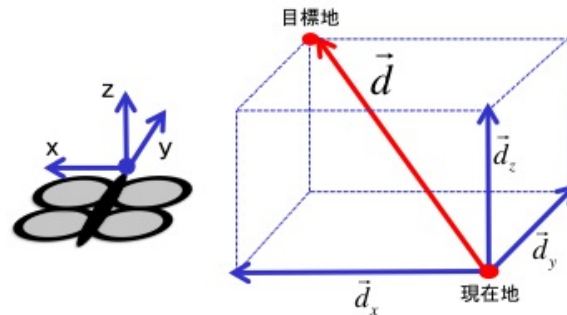


図 4.3 方向ベクトルの分解

向の速度とスケールが異なるため、定数 a を乗算することで各軸の速度を等しくする必要がある。

$$V_x = \frac{d_x}{|\vec{d}|} V$$

$$V_y = \frac{d_y}{|\vec{d}|} V$$

$$V_z = \frac{d_z}{|\vec{d}|} aV$$

4.1.6 デッドレコニング

AR.Drone には加速度センサが搭載されており、各軸の現在速度を航行データから得られるため、デッドレコニング機能を実装した。デッドレコニングとは相対的な自己位置推定のことであり、自身の位置を空間座標として逐次算出し求めることができる。前後方向と左右方向の速度データを逐次積分し、高度データと合わせることで現在位置を 30 ミリ秒間隔で空間座標として算出するよう処理を行った。デッドレコニングにはセンサの観測地に含まれている雑音や、サンプリング間隔の問題から、移動距離に比例して誤差が大きくなる問題がある。しかし提案手法において、得られた座標は移動方向の決定の際、部分的に利用しているため、累積誤差は無視できるほど小さく、システムに影響を与えない。最初の離陸点を原点として得られた空間座標データは、csv 形式でアプリケーションの終了と同時に記録されるように設定し、後に航行軌跡を参照することが可能である。図 4.4 に離陸、前進、着陸の一連動作を行った際のデッドレコニングデータをプロットしたグラフを示す。

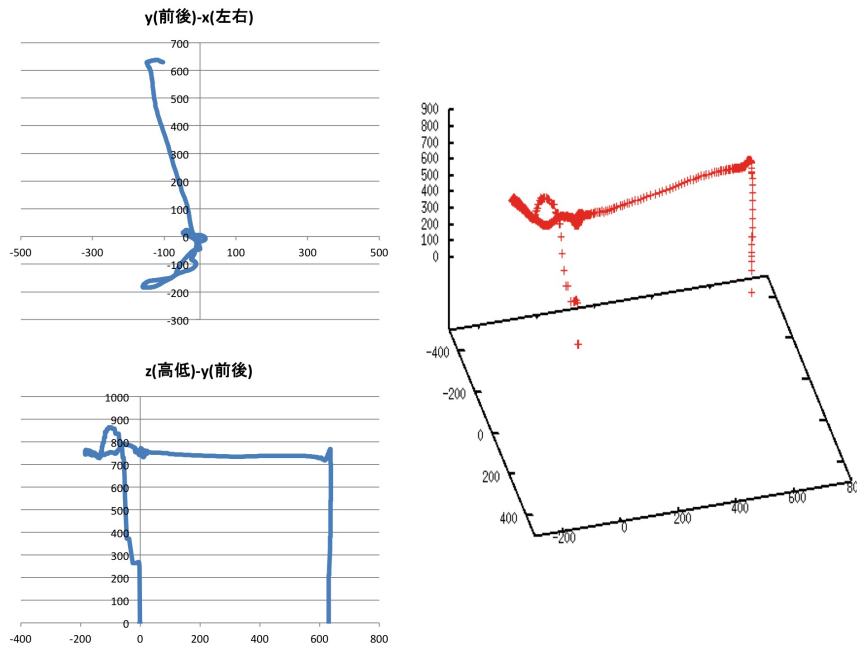


図 4.4 デッドレコニング

4.1.7 プロトタイプ(ノート PC)の考察

ノート PC 上に実装したプロトタイプは日本バーチャルリアリティ学会第 16 回シンポジウムや岐阜県立岐阜高校にてデモンストレーションを行っている。その際に得られたフィードバックを以下に挙げ、課題及び問題点を考察する。

- 2次元のパネルを表示する発想がユニークで操作が面白い
- レバーやたくさんのボタンを考えなくていいのは楽である
- 奥行き方向の操作が難しい
- パソコンでは難しいが、スマホやタブレットでできるようになるとカメラの位置に融通がきいて操作しやすいと思う
- AR を利用して現在位置の着地点などが表示されると分かりやすい
- 上下左右に限定した斜め移動は非常に操作しやすい
- AR の画像がぶれて操作しにくい

- どのようなアプリケーションが考えられるのか気になる

多くの意見として、発想、及び操作自体が面白いという感想が得られた。単純にラジコンヘリを操作する面白さと、AR を利用した2次元平面を用いるという操作の新鮮さから得られた意見だと考えられる。仮想平面の任意の位置をクリックするだけで移動が可能なることから、簡単な行程で目標位置への移動が実現できる、という感想が得られ、想定していた提案手法の利点が指摘された。その反面、画面の奥行き方向に対して、移動が困難であるという意見が得られた。これは仮想平面を回転させて奥行き方向の移動を実現する際、平面がどの方向に傾いているのか、どの程度傾いているのかが視覚的に分かりにくく、また平面の傾きが90度に近づく程、軌跡の記述範囲が狭まってしまうことに起因すると考えられる。ノートPCに実装した本プロトタイプでは操作が複雑である、と指摘を受けたが、これはノートPCとWebカメラとマウスを使用している都合上、同時に多くのデバイスを操作しなければならないため、ユーザが混乱しやすく、またノートPCは基本的に固定されているため、移動するAR.Droneに追従することができず、全体的に操作が困難になったことが原因と考えられる。また、ARの利用に関しては、高度データの反映による着陸箇所表示の他に、速度データに対応し大きさや色に変化する矢印の表示などが今後追加する機能として考えられる。今回はマーカの矩形を認識するARToolKitをARの認識に利用していたので、空中に浮かぶマーカは大きく揺れて認識が安定せず、仮想平面の表示がぶれてしまうことが多かった。

以上のことから、平面の傾き具合を視覚的に表示すること、ユーザの視点と仮想平面が平行になってしまう場合の対処、ARマーカの認識精度が課題として挙げられる。本プロトタイプにて仮想平面上を移動する操作システムが実現できたので、得られた意見を反映し、続いてタブレットPCへの実装を試みた。

4.2 タブレットPCへの実装

本プロトタイプでは、タッチスクリーンを用いて仮想平面の回転操作、拡大縮小の調節や移動経路の指示を行う。ノートPCに実装したプロトタイプと大きく異なる点を以下に簡単に挙げる。次節から開発環境、タッチスクリーンの操作方法を含めたインターフェースの説明、AR.Droneの曲線移動と速度制御の方法についてそれぞれ述べる。

- プラットフォームがタブレットになったことでWebカメラ、PC、マウスの機能が1つにまとまった。

- 移動軌跡を自由曲線で記述することが可能となり、任意の経路で移動指示が行えるようになった。
- 想定していた旋回、離着陸、動作の中断操作を実装した。
- その他操作を分かり易くするインタフェースの変更を行った。詳細は後述する。

4.2.1 開発環境

タブレット PC に実装した開発環境を表 4.3 に記す。

タブレット PC	iPad Air iOS7 (Apple 社)
プログラミング言語	Objective-C
AR 環境	Vuforia (Qualcomm) ³

表 4.3 タブレット開発環境

AR 環境はモバイル端末上で利用し易く、フリーで配布されている Vuforia を採用している。ARToolKit はマーカの矩形を検出しているため、飛行により安定しない位置にあるマーカは検出されにくく仮想平面の表示のぶれが多かった。Vuforia はマーカの特徴点を算出しているため、マーカの一部が多少映らなくても仮想平面を比較的安定してレンダリング可能である。

4.2.2 操作インタフェース

AR.Drone の周囲 4 方向に画像マーカを搭載し、いずれかのマーカをタブレット PC 搭載のカメラで検出、半透明な仮想平面を重畳する。平面は奥行き知覚のためにグリッド線を備えている。仮想平面は同様に、自由に回転、拡大縮小が可能であり、本実装では経路の記述は自由曲線で指定することができる。以下にタッチスクリーン上で行う操作方法を述べる。

仮想平面の回転操作は、タッチスクリーンを 0.4 秒以上ロングタップした後に指をスクリーンから離さずドラッグすることで行う。0.4 秒という値は iOS のロングタップと認識されるデフォルトの値である。ロングタップを認識した際はユーザフィードバックとして仮想平面の色を黄色から赤にするよう設定している。また今後断らない限り、操作に使用する指は 1 本で行うこととする。回転操作を行う際、2 軸以上の同時回転をドラッグ操作で

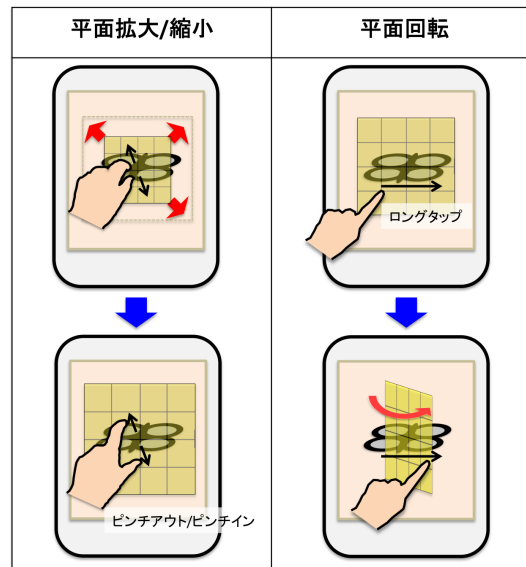


図 4.5 仮想平面の操作方法

行うことが困難なため、回転軸を1軸に限定しており、画面に表示されたボタン操作で鉛直方向の軸か平行方向の軸かを任意に変更できる。平面の拡大縮小操作はタッチスクリーンをピンチアウトすることで拡大、ピンチインすることで縮小を行う。仮想平面操作の様子を図4.5に示す。

平面の大きさと向きを決定した後、平面上に移動させたい移動軌跡を指定する。移動指示は見やすさのため、赤く表示したマーカの中心点から画面をドラッグすることで行う。画面をドラッグしている間は飛行ロボットの中心からタッチしている箇所に移動軌跡が表示され、画面から指を離すことによりAR.Droneは軌跡に従って移動を開始する。移動軌跡は中心点からのみ描けるようにしている。視線方向に対して平行にAR.Droneを移動させたい場合、仮想平面がスクリーンに対して垂直な位置関係になるため、軌跡の記述や仮想平面に対する操作が難しくなることが想定される。このような場合には、ユーザ自身がAR.Droneの側面方向に移動するなど、位置を適宜調節することでその問題を回避できる。

AR.Droneが着陸している状態で、仮想平面上に上方向の移動軌跡を描くと離陸動作を行う。また離陸している状態で仮想平面上に一定高度以下まで移動軌跡を描くと着陸動作を行う。着陸を行う高度は任意に指定できるものとし、図4.6に示すように仮想平面上に視覚的に表示する。

AR.Droneの旋回操作はタッチスクリーンを2本指でドラッグすることで指示する。指をタッチした位置から画面右方向にドラッグすることで左旋回、左方向にドラッグするこ



図 4.6 着陸を行う高度の表示

とで右旋回を行う。旋回する速度は現状 34.3 度/秒で一定であり、指をスクリーンから離れた時点で旋回はストップする。また AR マーカは AR.Drone の周囲 4 面に搭載しているため、AR.Drone が旋回しても、継続して仮想平面による操作が継続できる。

動作の中断はスクリーンをダブルタップすることで行う。移動経路を決定し、移動を行っている最中にダブルタップすることで、その場で移動を中断し、ホバリングを行う。またホバリングしている状態でスクリーンをダブルタップすると、仮想平面の大きさや向きがリセットされる。

4.2.3 AR.Drone の制御

4.1.5 節と同様に、AR.Drone が移動している最中は Move モード、移動せずにホバリングしている状態を Stay モードと定めている。しかし本プロトタイプではノート PC 上での実装とは異なり、曲線移動を実装している。以下に移動プロセスを述べる。

前述の通り、本プロトタイプは仮想平面上に自由曲線を記述できる。ユーザフィードバックとして軌跡は曲線的に表示されるが、実際は曲線のある程度の距離間隔で配置された点に分解している。そこで図 4.7 に記すように、原点 P_0 は配列 0、次の点 P_1 は配列 1、 P_2 は配列 2、と各地点の空間座標はマーカ座標系でそれぞれ配列に格納されている、現在地の座標から次の配列の座標との差分をとり、移動方向の単位ベクトルとベクトルの大きさを計算する。4.1.5 節と同様に得られた単位ベクトルを AR.Drone 機体の前後、左右、上下各軸の成分に分解し、それぞれの移動速度を決定して AR.Drone に P_0 から P_1 へ移動命令を送る。実際は、記述が終了し指を離れた瞬間に Stay モードから Move モードに切り替わり、移動が開始する。移動している最中、ベクトルの大きさとデッドレコニングにより求

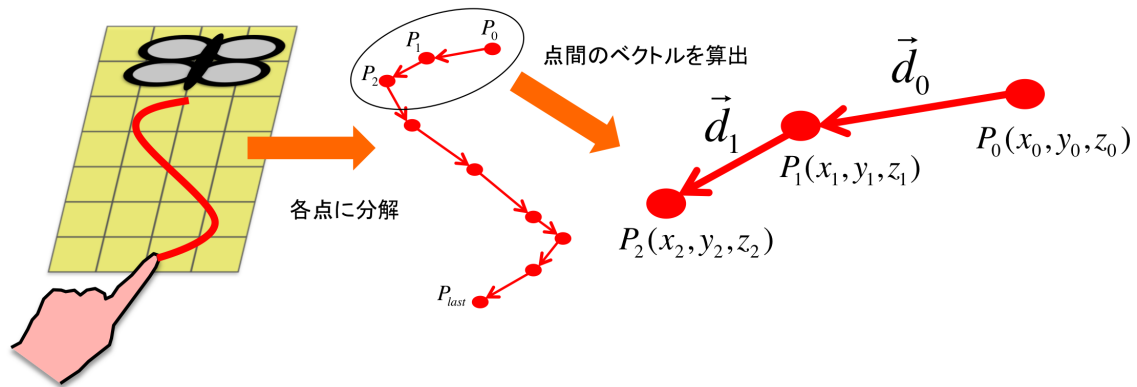


図 4.7 自由曲線配列

められた実際の移動距離は 1/30 秒ごとに比較され、移動距離が上回った時点で P_1 に移動が完了したと判定する。そして P_1 を現在地とし、次の P_2 に対して同様に計算をし、移動を行う。以上のプロセスを現在地点 P_0 から配列の終点 P_{last} まで順々に繰り返す。擬似的な曲線移動を実現している。終点まで移動を完了したら、Move モードから Stay モードに切り替わり、停止するよう処理している。

軌跡記述速度と移動速度の対応

本プロトタイプではロボット移動操作の際に、軌跡の記述速度とロボットの移動速度を対応させるよう実装した。以下に制御方法について述べる。

前述の通り、仮想平面上の移動軌跡描写は一定間隔でサンプリングし、各点に分解している。よって軌跡を記述する際、記述速度によって各点の距離は変化する。そこで点間の距離に応じて速度を変化するように制御を行った。各方向の速度を求める式を以下に示す。 P は現在移動している点間の距離を示し、 M は速度を変化させる点間の最大距離をそれぞれ示す。 $P > M$ の際に P には M の値を格納する。 M の値は可変であるが、現在は 80 ミリメートルに設定している。サンプリングは 1 秒間に 30 回行っているため、仮想平面上を 2.4 メートル毎秒の速度で記述することにより最大速度 V_{max} となる。 V_{max} の値は操作環境に合わせて任意に変更可能である。

$$V_x = \frac{d_x}{|\vec{d}|} V_{max} \times \frac{P}{M}$$

$$V_y = \frac{d_y}{|\vec{d}|} V_{max} \times \frac{P}{M}$$

$$V_z = \frac{d_z}{|d|} a V_{max} \times \frac{P}{M}$$

第5章 評価実験

提案手法による飛行ロボット操作性評価のため、2種類の実験を行った。まず、予備実験として、提案手法を用いて飛行ロボットを操作した際、特定の操作タスクを遂行するまでの時間を測定した。離陸から設定した場所への移動、旋回を行い、着陸までに要する操作時間の内訳を明らかにすることで、提案手法のどの操作により多く時間がかかってしまうのか、短縮は可能かを検証した。また、実験を通して AR.Drone の2種類の移動速度制御について比較、検証を行った。続いて、飛行ロボットの一般的な操作手法であるジョイスティックタイプのコントローラと提案手法を比較評価するために、空間上の現在位置から目標位置までの移動にかかる時間をそれぞれの操作手法で測定した。また、実験後に行ったアンケートによって、提案手法とジョイスティックコントローラとの主観的な比較を行い、それぞれの操作性に有意差があるかを検証した。次節以降で、各実験の詳細について述べる。

5.1 予備実験：特定の操作タスクを与えた際の所要時間明示と速度制御比較

5.1.1 予備実験：目的と内容

コントローラの操作性を評価する1つの尺度として、タスクにかかる操作時間が考えられる。そこで評価実験に用いるプロトタイプにおいて、特定のタスクを与えた際にかかる AR.Drone の各操作に要する時間を明確にし、検証する。また提案した速度制御方式について比較、検証するために AR.Drone の移動速度を一定にした場合と、軌跡の記述速度と移動速度を対応させた場合で測定をそれぞれ行う。短縮が可能な項目に関しては対処を施し、また速度対応を行うべきかを検証した上で、評価実験に臨む。被験者が AR.Drone を操作する空間では図 5.1 に示すように、1番から3番までの離着陸箇所を設定している。本実験では離陸から、各番号間の移動、着陸までを一連の操作とし、時間計測を行う。計測する時間はそれぞれ、仮想平面の拡大縮小操作時間、仮想平面回転操作時間、移動軌跡描写時間、旋回時間、移動時間、何も操作していない待機時間である。ただし、AR マーカが認識から外れている時間は待機時間に含めない。また、離陸から着陸までの間に必ず旋

回動作を行うように設定しており、1-2番間、2-3番間を移動する際は90度旋回、3-1番間は180度の旋回を行う。それぞれの番号は2番を直角とする直角二等辺三角形の各頂点に配置しており、1-2番間と2-3番間は1.3メートル間隔、3-1番間は1.85メートル間隔である。番号間それぞれの位置関係は操作している被験者から見て左右方向、奥行き方向、左右・奥行きの両方向である。提案手法では仮想平面を画面上に表示させるため、ARマーカを用いているが、今回作成したマーカの認識は、カメラから約1.8メートル程離れると認識しなくなってしまう。そのため、それぞれの番号はマーカの認識が可能な位置に配置している。試行回数は各番号間両方向の移動を5回ずつとする。1-2番間の計測を例とすると、1番から2番への移動を5回、2番から1番へ移動を5回、よって1-2番間の移動は合計10回となる。それらを軌跡の記述速度と移動速度を対応させた場合と対応させていない一定速度の場合で別々に計測する。一定速度の場合は85センチメートル毎秒、速度対応の場合は0～117センチメートル毎秒の範囲で記述スピードに対応し変化する。前者は、常に大きい速度では仮想平面に描いた軌跡を慣性の影響により外れてしまうため、正しく制御できる速度に設定している。今回は操作ミスを極力減らした状態で計測を行うため、ある程度提案手法の操作に慣れた開発者本人(26歳男性)が実験している。また、参考として飛行ロボットの一般的な操作方法であるジョイスティックタイプのコントローラでも同様の実験を行い、操作にかかる時間を計測する。使用したジョイスティックコントローラは、AR.Droneの開発元であるParrot社からフリーウェアとして配信されているアプリ“AR Flight”を使用し、タブレットPC(iPad)で操作を行う。AR Flightの操作画面を図5.2に示す。左右にそれぞれ配置されたジョイスティックを仮想的にした半透明のパッドと、中央下に配置されたボタンによって主に操作を行い、左のパッドは機体に対して前後左右方向の移動操作に対応し、右のパッドの上下方向は機体の上昇下降操作、左右方向は機体の旋回操作に対応している。着陸状態で中央下のボタンを押すことで離陸操作、飛行状態でボタンを押すことで着陸操作を行う。移動速度はスティックを倒した移動量に対応しており、最高速度は117センチメートル毎秒である。速度対応を行った提案手法の速度設定はこの値と合わせている。

5.1.2 予備実験：結果と考察

記述速度と移動速度の対応なし、対応あり、ジョイスティックコントローラそれぞれの操作でタスク遂行にかかった所要時間と、各操作時間の内訳を図5.3に示す。それぞれのグラフは各番号間の移動別に集計を行っている。システム全体を通して、仮想平面の回転・拡大

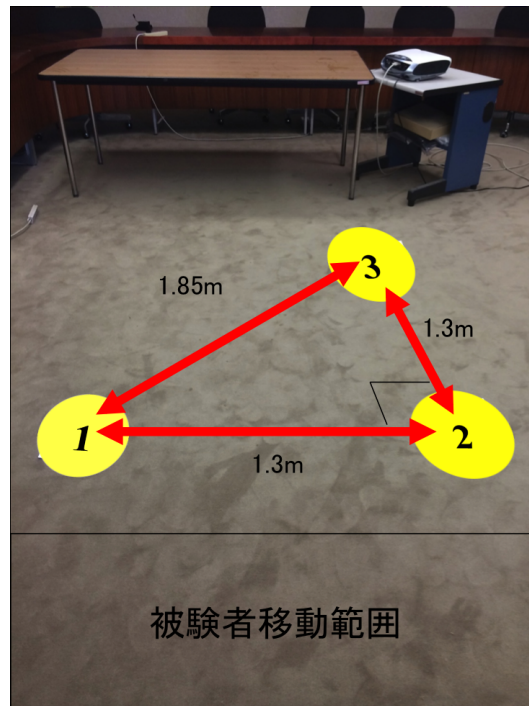


図 5.1 予備実験：状況設定



図 5.2 AR Flight 操作画面

縮小や移動軌跡描写など，ユーザが端末を操作している時間は速度対応なしは各試行平均して1.74秒，全体の9.1%であり，対応ありは平均して1.20秒，全体の7.7%であった．すなわち，全時間に対して端末を操作している時間の割合は小さいという結果になった．その

反面、何も操作を行わない待機状態は対応なしの場合、平均して4.76秒、全体の24.8%であり、対応ありは平均して4.00秒、全体の25.7%であった。大きな割合を占めているものの、何も操作していない時間では主に仮想平面の角度や移動経路を考えており、今回の実験設定では操作に慣れた被験者を採用しているため、大幅に短縮することは難しい。続いて移動・旋回時間に注目する。1-2番間、2-3番間の旋回時間と3-1番間の旋回時間は約2倍程度の差が見られるが、これは旋回角度に比例しているからであると考えられる。旋回速度はジョイスティック操作と比べてもほぼ等しく、短縮する必要性はあまりない。また、移動時間に関しては全体を通して、1-2番間、2-3番間の移動に比べ、3-1番間の移動に多くの時間を要するが、これはそれぞれの番号間の距離に比例していると考えられる。また速度対応ありの場合は、対応なしの場合と比較して22.0%減少している。移動操作は、基本的に直線的な移動と、目的地付近で行う微調整を行う移動の2種類を基本的にを行う。対応なしの場合はこれらを一定速度で行うが、対応ありの場合は直線移動の際に速度を大きく、微調整移動の際に速度を小さく使い分ける。微調整移動では速度が小さくても移動距離も短いため、対応なしと比較しても大きな差は生じない。そのため全体を通して平均速度が上がり、対応ありの場合は操作時間が短縮されたと考えられる。以上の結果から、評価実験では記述速度と移動速度を対応させた制御方式を採用する。

参考として、ジョイスティックの操作時間を見ると、提案手法の両速度制御方式に対して、少ない時間でタスクを遂行する傾向がある。提案手法では移動軌跡の記述操作とAR.Droneの移動に時間差が生じることから、操作と移動を同時に行うジョイスティックと比較して、より多くの時間を要すると考えられる。そこで、提案手法において、AR.Droneが実際に移動・旋回した時間のみを抜き出したデータと、ジョイスティックの操作時間を併せて図5.4に示す。ジョイスティックコントローラでは操作と移動が同時であることから、常に移動していると仮定すると、最高速度が提案手法の速度対応ありと等しいジョイスティック操作は、ほぼ等しい所要時間になるはずである。しかし実際は提案手法と比較して長くかかる。これは、ジョイスティックコントローラではスティックを倒した移動量が仮想的であるため速度の調節が困難であり、微調整に時間がかかってしまうことと、旋回によって絶えずユーザとの方向関係が変化し、移動方向を間違えてしまうことで余分に時間がかかるためであると考えられる。

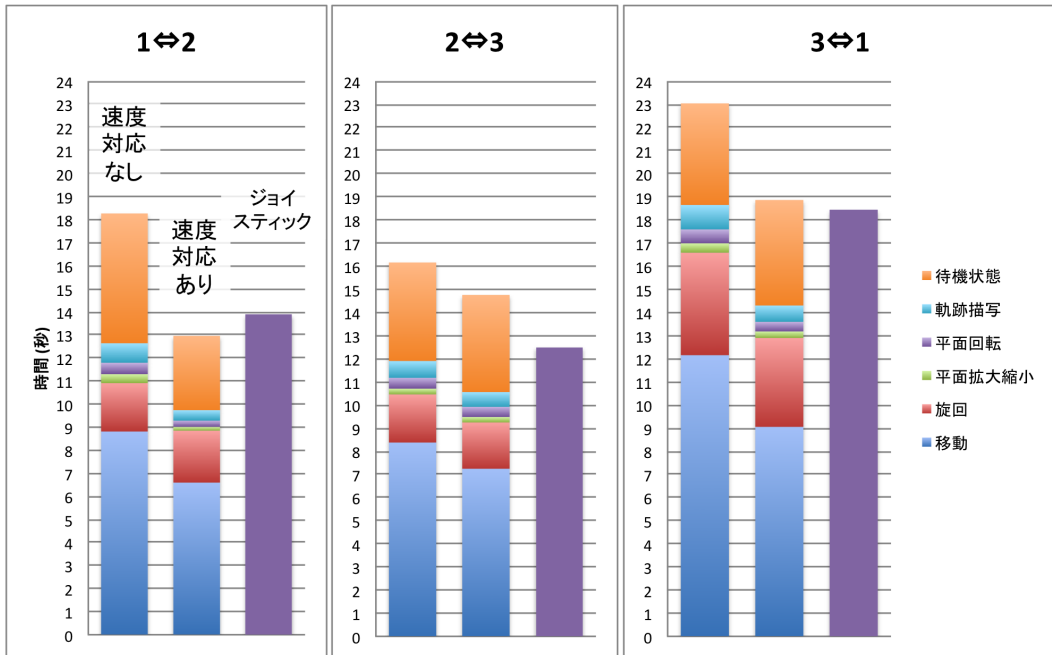


図 5.3 予備実験：各番号間移動の所要時間

5.2 評価実験：提案手法とジョイスティックコントローラとの操作性比較評価

5.2.1 評価実験：目的と内容

提案手法の操作性評価のため、指定の移動操作に対し、提案手法とジョイスティックコントローラとの比較を行った。今回の実験では特にロボットの前後左右方向と上下方向の同時移動、つまり空間上の斜め移動について、検証を行う。今回比較を行うジョイスティックコントローラは、5.1.1節に記した通り左パッドを用いてAR.Droneの前後左右移動を、右パッドを用いて上下移動と旋回動作を行う。そのため、上下軸と前後左右平面を同時に移動する斜め移動の際、両パッドを同時に操作する必要がある。両パッドを用いた操作は、習熟していないユーザには容易ではなく、斜め移動の際は片方のパッドを交互に操作することで達成することが予想される。しかし、提案手法においては平面上の記述によって2軸以上の斜め移動操作に差はなく、一直線に経路を指定し、移動を行うことが可能であり、容易に目的地に到達できる。以上のことから2つの操作には時間差が生じると考えられる。本評価実験では斜め移動を行う状況設定においてその時間差を確認し、ジョイスティック操作と提案手法それぞれの操作を観察することで、その差を検証する。そして、実験後の

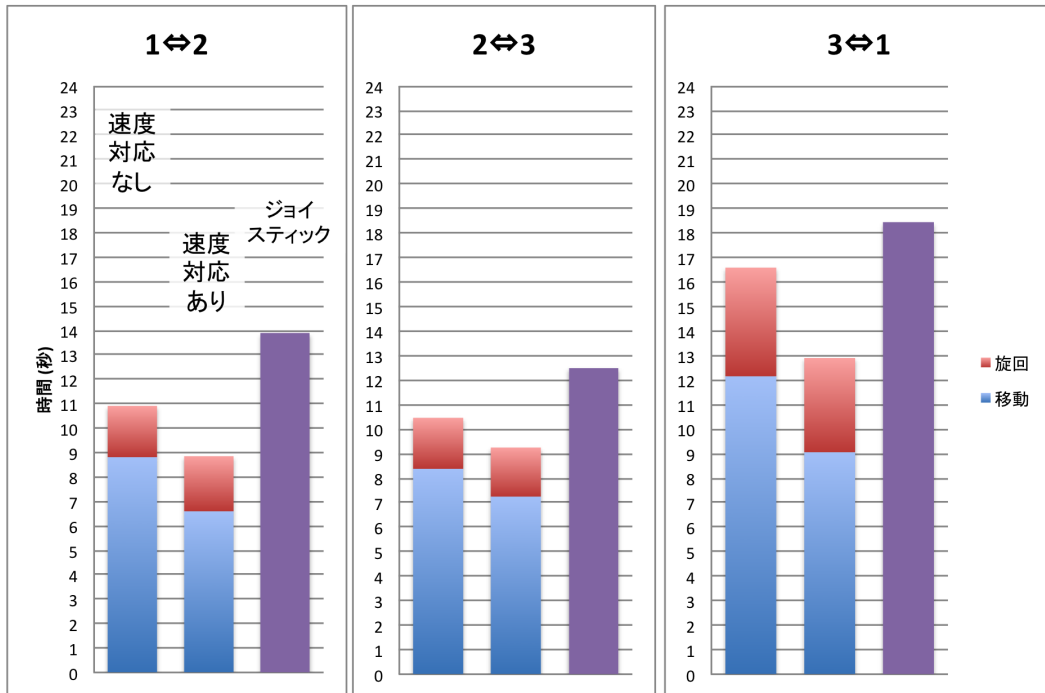


図 5.4 予備実験：移動・旋回操作のみの所要時間

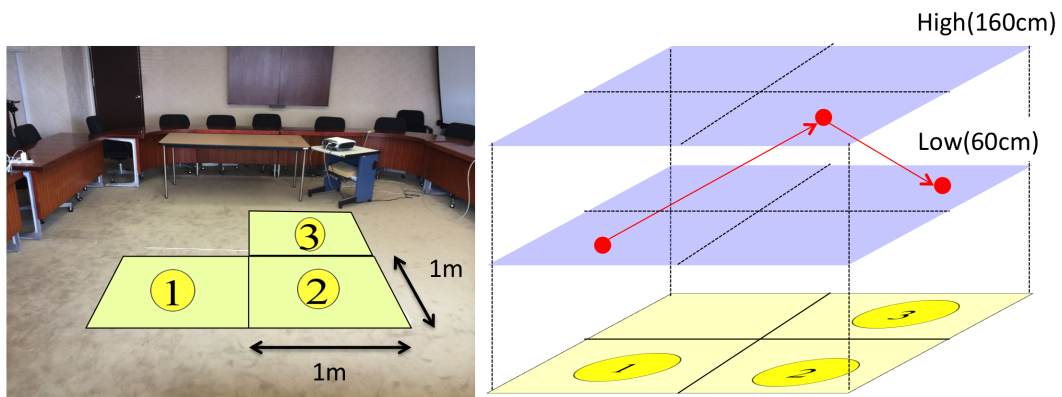


図 5.5 評価実験：状況設定

アンケートを通じて主観的に操作性を評価する。比較を行うジョイスティックコントローラは予備実験と同様に“AR Flight”アプリを利用する。

具体的には図 5.5 の状況において予備実験と同様に 3 つの箇所をそれぞれ移動していく。番号間の移動に加えて、今回の実験では床より約 160 センチメートルの高さを High、床より約 60 センチメートルの高さを Low として上下方向の位置も指定する。そこで、本実験の

移動パターンは 1Low → 2High, 1High → 2Low, 2Low → 3High, 2High → 3Low, 3Low → 1High, 3High → 1Low の合計 6 パターンとする (図 5.5 には例として 1Low → 2High, 2High → 3Low の移動経路を示している). 移動の初期位置は各番号の真上とし, 移動の完了は各番号を中心とした 1メートル四方の空間に AR.Drone がおさまっていることを条件とする. 判定は筆者が目測で行う. 比較は提案手法とジョイスティックコントローラについて行うが, ジョイスティック操作ではユーザと AR.Drone の向いている方向が一致している場合と, 一致していない場合 (ユーザに対して AR.Drone は左方向を向いている状態) の 2 種類, 提案手法と合わせて 3 種類測定し, 比較を行う. 被験者にはジョイスティック操作を行う際, 可能であれば斜め移動を行うように指示している. また, 提案手法操作の際に仮想平面を回転する軸の選択はあらかじめ指導している. 測定項目は, 各移動が完遂するまでの時間と, 移動方向のミス回数である. 後者に関しては実験風景をビデオに撮影し測定を行う. そして, 実験後に主観評価としてユーザにアンケートを行い, 7段階のリッカート尺度を用い評価する. 被験者は男性 8 名, 女性 1 名の計 9 名である. また実験前に提案手法, ジョイスティック両操作の練習時間を設けた. 被験者が十分と感じた時点で練習を終え, 最大 10 分間行った.

5.2.2 評価実験：結果と考察

各 6 パターンそれぞれの移動に要した時間を, 3 種類の操作手法それぞれに対して標準誤差と共に図 5.6 に示す. また, 提案手法の移動動作のみを抜き出した時間とジョイスティックの操作時間を比較した結果を標準誤差と共に図 5.7 に示す. 2 つの図には提案手法とジョイスティック方向一致, 不一致操作それぞれの結果に対して有意水準を 0.05 とした両側 t 検定の結果も併せて示している. 全体の所要時間を見ると, ジョイスティック方向一致操作が平均 6.52 秒で一番短く, 次にジョイスティック方向不一致で平均 10.34 秒, そして一番時間を要する提案手法は平均 11.30 秒であった. 5.1.2 節にて行った考察と同様に, ユーザとロボットの方向が不一致な状態では方向感覚の齟齬から目的位置に直行することが困難であり, 方向一致の際と比べて時間がかかった. また, 提案手法とジョイスティック方向一致操作ではほとんどの移動パターンで有意差が生じ, 提案手法の方がより多くの時間を要するという結果を得た. 提案手法は操作と実際の移動に時間差が生じることに加え, 今回の実験では提案手法の操作はほぼ初見であることから, 慣れない操作に手間取ってしまったと予想できる. そして, インタフェース操作時間や待機時間を除き, 移動操作のみにかかった時間に注目すると, 要した時間は平均 2.72 秒と, ジョイスティック方向一致, 不

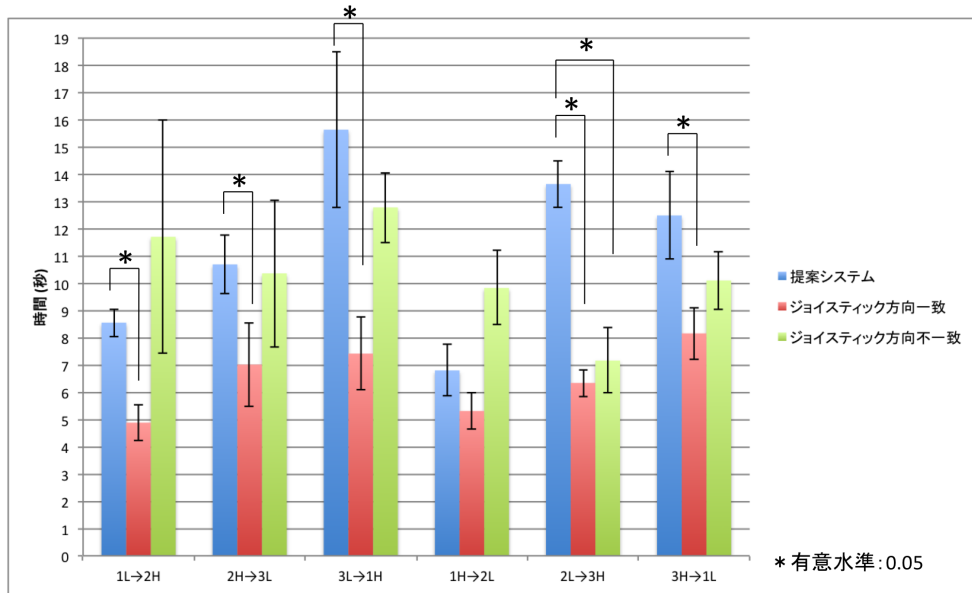


図 5.6 評価実験：所要時間

致操作それぞれのほぼ全ての移動パターンに対して有意差が生じ、短い結果となった。実際はジョイスティック操作の際も何も操作していない待機時間は存在するが、方向一致操作の場合は、実験前に十分な練習を行っており、また方向感覚の齟齬がない操作なため、ほぼ操作している時間と考えられる。以上のことから提案手法では斜め移動を行う際、目的地にほぼ直線的に移動を行っていることに対し、ジョイスティック操作では複数の操作を組み合わせた擬似的な斜め移動を行っており、提案手法ではより効率良く指定の位置に移動していると考えられる。また、実験の様子を撮影したビデオカメラの映像から、ジョイスティック操作の様子を観察すると、目的地へ一直線に斜め移動を行う試行と、複数の移動操作によって擬似的に斜め移動を行う試行がそれぞれ見られた。それぞれの移動方式を行う割合は個人差があるが、全ての被験者が擬似的な斜め移動を一度以上行っており、全試行に対して方向一致の際は 50.0%、方向不一致の際は 64.8%の割合で複数操作による擬似的な斜め移動を行っていることが確認された。方向不一致の方がより割合が高いのは、方向感覚が混乱している状態で2つのジョイスティックを同時に操作することが困難なためと考えられる。また、今回の実験では被験者が次に移動する箇所がランダムではなく予想可能なため、実際の計測前に移動方向を考える時間が生じてしまった。そのため、移動指示をランダムに行った場合、方向不一致操作に関してこの割合はより大きくなることが予想される。

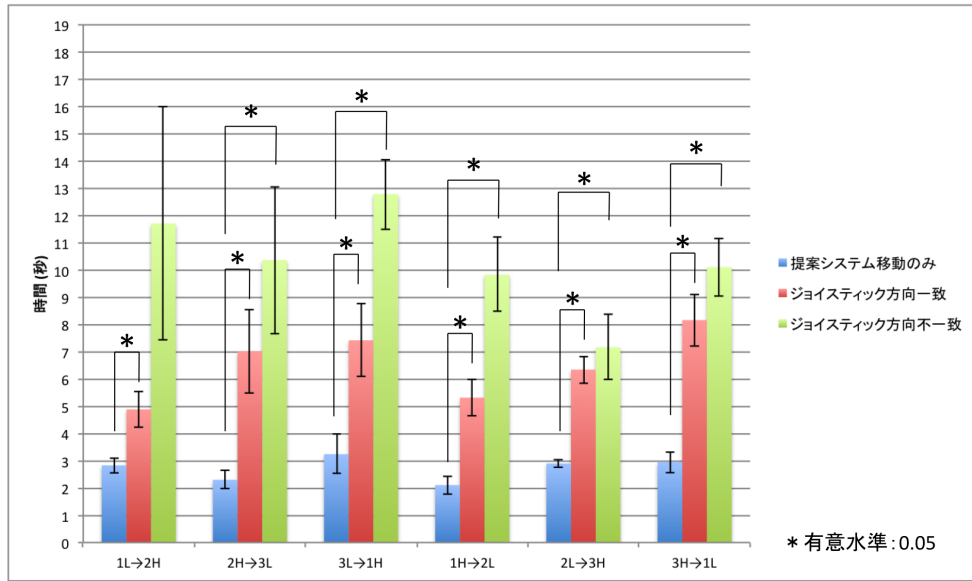


図 5.7 評価実験：移動操作のみの所要時間

続いてジョイスティックの方向不一致操作時の一連の実験中に生じた操作ミス回数を表 5.1 に記す。本実験における操作ミスとは、目的の方向に対して別の方向に移動を行った操作と定義する。被験者間でばらつきが大きく、全くミスをしない被験者もいれば、実験を通して最大 11 回の操作ミスを生じてしまう被験者もいた。後に聞くと、多くミスをしてしまう被験者は普段テレビゲームなどを全く行わず、手持ちのコントローラを操作する習慣がほぼないことが分かった。このことから、ゲームなど仮想的にでもジョイスティックコントローラを用いる習慣の有無によって、飛行ロボットの操作に影響が出ると考えられる。また、操作ミスが 0 回という被験者の多くは、移動の開始合図と共に数秒考えてから操作を始める、という傾向が見られた。これは図 5.6 のジョイスティックの方向一致と不一致の所要時間の差に表れていると考えられる。

被験者番号	1	2	3	4	5	6	7	8	9
操作ミス回数 (回)	11	0	1	1	3	0	0	1	2

表 5.1 評価実験時の操作ミス回数

次に提案手法操作の各操作の内訳を図 5.8 に示す。移動時間が短い反面、操作時間の大半を占めているのは何もしていない待機時間である。これはマーカの認識に手間取っている時間やユーザが仮想平面の向きや大きさを考えている時間であり、ほぼ初回の操作であ

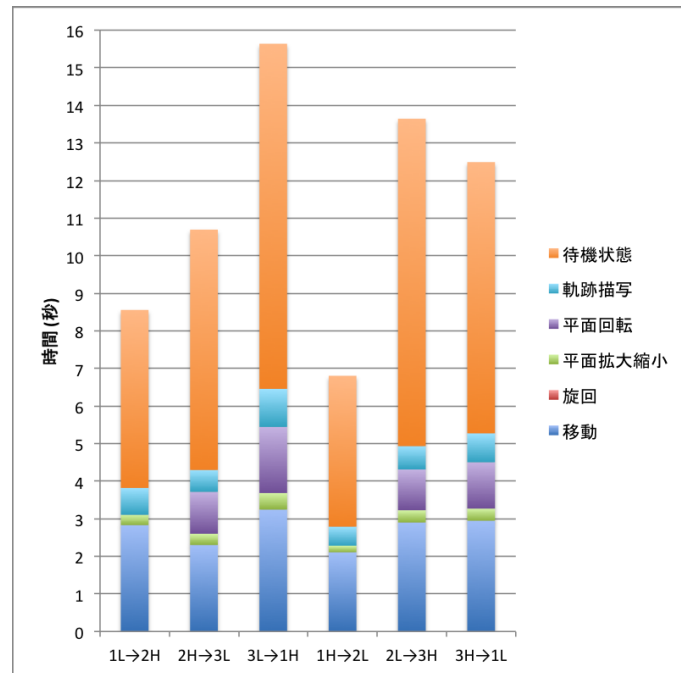


図 5.8 評価実験：各操作時間の内訳

る本実験では時間がかかってしまうことが考えられる。仮想平面の向きや大きさを考えている時間が占める、というのは実際の仮想平面操作、記述操作時間の長さにも表れていると予想されるため、図 5.9 に記したように待機時間とインタフェース操作 (仮想平面拡大縮小+仮想平面回転+軌跡記述) 時間との相関を取った。結果は相関係数 $0.76 > 0.7$ という強い相関が確認され、先ほどの考察が正しいことを裏付けた。先ほど述べた通り、今回の被験者は提案手法を操作するにあたってほぼ初回であることから、全体的に所要時間が多くかかったと予想される。そこで十分提案手法の操作に慣れたユーザと考えられる開発者も同様に実験を行い、各操作の内訳を明らかにした結果を図 5.10 に示す。比較すると、移動時間は 25.6%、インタフェース操作時間は 20.0%、待機時間は 36.0%それぞれ短縮された。よって、全体を通して操作の慣れにより操作時間の大幅な短縮が見込まれ、待機時間は特にその影響が大きいと考えられる。

最後にアンケートの結果を標準誤差と共に図 5.11 に示す。図には提案手法とジョイスティック方向一致、不一致それぞれの結果に対して有意水準を 0.05 とした両側 t 検定の結果も併せて示している。評価は 7 段階で行われ、全ての項目においてポイントが高い程操作性は良い評価となる。「操作しやすいと感じた」「自信を持って操作できた」「効率良く目的地に移動できた」「ロボットを想像通りに動かすことができた」という項目はほぼ同様の結

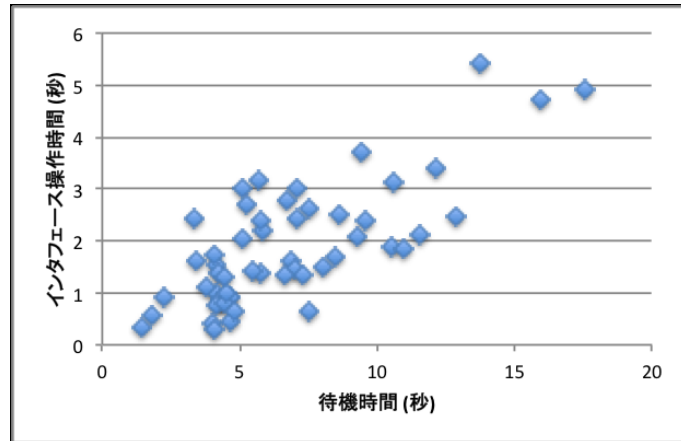


図 5.9 評価実験：待機時間とインタフェース操作時間の相関

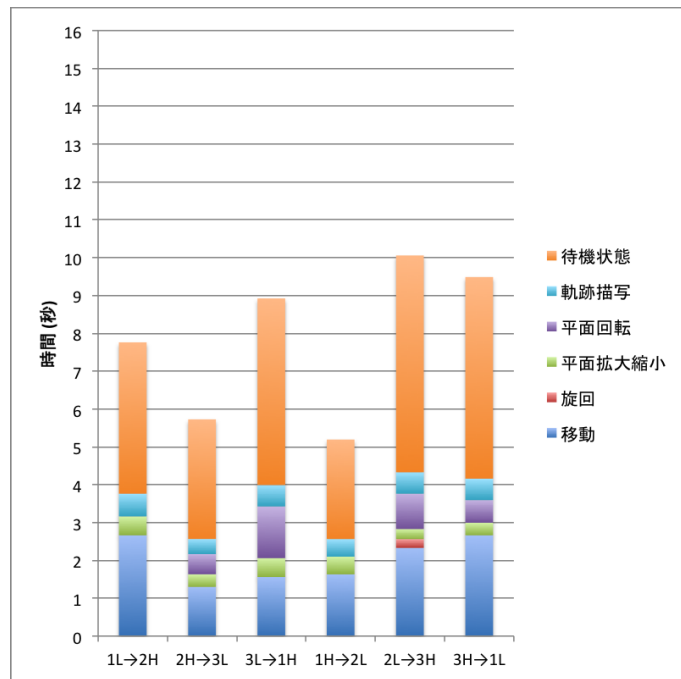


図 5.10 評価実験：操作に十分慣れたユーザの各操作時間の内訳

果を示しており、提案手法操作に対してジョイスティック方向一致操作とは有意差は生じず、方向不一致操作とは有意差が生じ、提案手法の方が優れている結果となった。このことから、ユーザとロボットの位置関係に依らず直感的に、また容易に操作できることが見て取れる。また「斜め移動は容易であった」という項目はジョイスティック方向一致、不一致とも優れている方に有意差が生じ、先述の時間比較を行った結果と共に提案手法の特

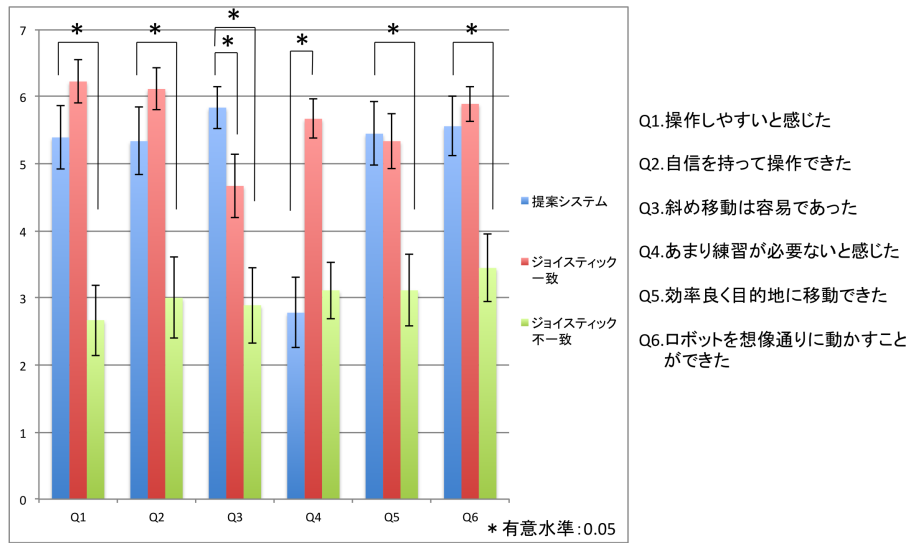


図 5.11 評価実験：アンケート結果

長が裏付けられる。しかしその反面「あまり練習が必要ないと感じた」という項目に関してはジョイスティック方向一致と有意差が生じ、提案手法の方が劣った結果となった。これは全く新しい飛行ロボットの操作システムである提案手法に対し、仮想的であってもテレビゲーム等でジョイスティック操作には慣れ親しんでいることに起因すると考えられる。

第6章 考察

本章では実験結果を踏まえて提案手法の特徴と課題について考察する。

6.1 提案手法の特徴

本提案手法は、3次元空間を移動するロボットの操作を、2次元の操作に落とすことで、比較的容易な操作を実現する操作手法である。5節の評価実験によって、ユーザとロボットの方向関係に左右されず、直感的且つ容易に操作できることを示した。またロボットの前後左右方向と上下方向の同時移動、すなわち空間上の斜め移動において、ジョイスティックコントローラを用いた場合は複数の移動指示を同時に行う必要があり困難な操作になるが、提案手法では現在地と目的地を通るように仮想平面の向きを指定していれば、全方向に対して指示する操作は変わらず、経路の記述によって容易に目的地に移動できることを示した。本節では上記に加えて、提案手法の特徴について考察する。

6.1.1 操作視点

飛行ロボットに限らず、ロボット操作を行う場合は2種類の操作視点が考えられる。1つは車など操作する対象に直接乗り込む、もしくはロボットに搭載されたカメラの映像をユーザに投影することによる一人称視点の操作である。遠隔操作の場合は、ユーザから遠く離れた場所においてもロボット前面の映像を頼りに操作を行うことが可能であるが、視野角の問題から、ロボット周囲の状況を把握しにくく、障害物を感知しにくい問題がある。もう1つは操作対象のロボットを俯瞰で見る三人称視点の操作である。一人称視点では全く見えない後方も含めて、ロボット周囲の状況を把握することができる操作視点であり、その有効性が報告されている [37]。しかし、三人称視点において操作を行いたい方向と、実際に操作を指示する方向は一致しないことが多く、ロボットの現在向いている方向を常に意識して操作を行わなければならない。また俯瞰の映像は部屋に設置したカメラ、もしくはロボット本体に設置したパノラマカメラから得る状況が多く、固定されたカメラでは動的

で広い環境の視点は得られない。その問題を解決するために俯瞰映像を動的に得る浮遊体を用いた三人称視点の生成手法が提案されているが [38, 39], 操作インタフェースとは別に特殊な装置が必要となり, 加えて空間を移動するカメラは風等の外乱により, 安定した映像情報を提供できない。本提案手法は三人称視点による操作であるが, 操作インタフェースと俯瞰の映像を撮影するカメラが一体となっており, 動的に俯瞰映像を変化させながら, ロボットの周囲を把握し操作を行うことが可能である。

6.1.2 移動経路の指定

移動経路を指定する操作はユーザとロボットの方向関係に依らず直感的に操作できる利点がある。そして一般的な移動方向を指示するコントローラは, 指示した命令に対し, 実際に移動した量を常に目で見て確認し続ける必要があるが, 移動経路を指定する操作は移動の始点と経路, 終点が視覚的にフィードバックされるため, ロボットの移動量を見積もることが可能である。本提案手法では, 経路を描く平面を仮想的に用意することで, 何もない空中であっても経路を指定することが可能となった。これにより3次元空間を移動する飛行ロボットであっても, 仮想平面の向きを任意に決定することで, 直感的かつ自由に操作を行うことができる。3次元的な操作は従来複数のレバーやボタン等を組み合わせることで実現されており, 2次元平面を移動するロボットの操作に比べて複雑なものであった。しかし提案手法では2次元平面の移動操作の簡易さを保ちつつ, 3次元空間に移動を行うロボットを操作できるという点が大きな特徴である。また, 2.3節にて記した既存手法では, ロボットの位置情報を固定カメラからの映像によって逐次取得することで, 指定した経路とのズレを修正しながら移動を行っているため, ロボットを視認できなくなると操作(移動)できないシステムだったが, 提案手法では移動ルートの指定が完了すれば, 現在位置や目標位置の情報はデッドレコニングによって映像情報を用いず算出しているため, 障害物によって一時的にロボットをカメラで捉えられなかったとしても移動が可能である。

6.2 課題

本節では提案手法を利用する上での課題について考察する。

6.2.1 仮想平面と実物体の位置関係

2次元のスクリーンで AR.Drone を操作する場合、画面の奥行き方向が知覚しにくく、ユーザの視線に対して前後方向の操作が困難になる場合がある。そこで仮想平面をグリッド表示にする処理を行い、奥行き方向への知覚を補助している。しかし、実物体に対して仮想平面が強制的に手前に表示されることから、移動させたい目標地への移動の際に、平面の向き設定や移動軌跡の描写を描いたりするのが困難な場合がある。この問題を解決するためには周囲の空間情報を何らかの方法で取得することで、仮想平面よりも手前に存在するオブジェクト上に、平面を表示させない隠面処理を行い、仮想平面と実物体の位置関係の把握を補助することが必要であると考えられる。周囲の空間情報を得るため、2台以上のカメラを使用し、2つの画像間の対応を求めることで、3次元の形状を復元する手法が研究されている [40]。リアルタイムで2つの画像から3次元復元を行う試みもなされ、1フレームあたり 37 ミリ秒で処理を完了できる報告がある [41]。今後の展望として、タブレット PC によって撮影した映像と、UAV に搭載されたカメラ映像にそれらの技術を適用することが考えられる。

6.2.2 移動開始のタイミング

移動制御において、実際に移動を開始する Move モードへの切り替えタイミングは移動経路の指定を終えた後である。実際の利用では、より迅速な移動操作を指示する場面も考えられるため、移動軌跡描画中にリアルタイムで目標位置に追従する移動など、動作開始のタイミングに関する操作性の検証を考えている。軌跡の描画中に移動を行う機能を実装するにあたって、現状の課題を述べる。現状ではロボットの移動と同時に仮想平面をディスプレイに固定するよう処理しているが、リアルタイムに平面上をポインティングし、移動操作を行う場合、ディスプレイは逐次動いてしまい、仮想平面も合わせて移動してしまう。タッチ位置に対してリアルタイムに追従する操作を行うためには、ロボットの現在位置と、タッチしている目標位置の座標差を取り、逐次移動命令にフィードバックすることで実現することができるが、ディスプレイの移動に合わせて逐次目標位置が変動してしまい、安定した動作は難しい。

6.2.3 飛行ロボットの移動制御

現在の移動制御では慣性の影響を考慮していないため、急激な曲線移動を行う場合、描いた軌跡を大きく外れてしまう。そこで、過去の速度データを参照し、逐次過去の速度を

打ち消すように速度を指定することで改善されうることが見込める。障害物を避けるように大きく曲線を描いて移動させる操作など、慣性の影響が無視できない操作も多く想定されるため、今後解決すべき課題である。

6.2.4 タッチスクリーン操作

4.2.2 節で述べたタッチスクリーン上での操作方法は、他にいくつかのアプローチが存在する。本節では被験者実験によって得られた意見を参考に、仮想平面を操作する上での現状の課題と改善案について考察する。

平面回転操作

仮想平面を回転させる操作は、回転させる軸を、ディスプレイ上の仮想的なボタンを操作することで選択した後、スクリーンをロングタップし、回転させたい向きにドラッグすることで行う。すなわちロングタップの有無によってドラッグ操作を移動経路の指定とするか平面回転とするかを切り替えている。ロングタップを行った判定として、仮想平面の色を変化する様にユーザへフィードバックしているが、この操作では、ミスタッチなどによって移動経路の指定と平面の回転操作が誤作動してしまうことがある。5.2 節の評価実験では、ディスプレイのサイズが大きいタブレット PC を使用しているため、一方の手でタブレットを持ち、もう一方の手でインタフェースを操作する被験者が多かった。その際に持っている手がディスプレイに触れてしまい、上記の様な操作ミスにつながるものが少なくなかった。

誤作動を防ぐために考えられる平面回転の操作例として、仮想平面の外枠 1 辺をタッチしドラッグすることで、その辺と平行でマーカの原点を通る直線を軸とする回転操作に割り当てる手法が考えられる。操作の様子を図 6.1 に示す。移動経路の指定との誤作動を防ぐことに加え、回転の軸を直感的に指定できること、ロングタップに必要な待機時間が少ないことが利点として考えられる。

平面拡大縮小操作

平面の拡大縮小操作は現在タッチスクリーンのピンチイン/アウト操作によって行う。しかし、5.2 節の評価実験では、本機能を使用しない場面も多く見られた。その理由は、ピンチイン/アウト操作に必要な 2 本指のタッチ操作が、同じく 2 本指のドラッグ操作によって

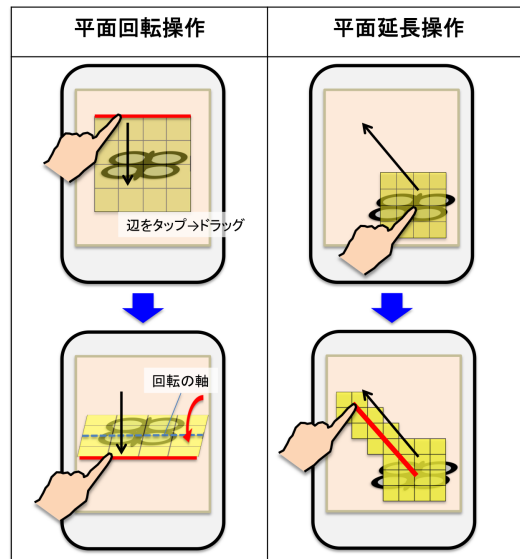


図 6.1 仮想平面操作の拡張

行われる旋回操作に誤認識してしまう場合が多かったためである。また、本来目的地まで平面を伸ばしたいが、目的地の反対側など不必要な部分まで平面が拡大され、画面を多く占有してしまい、平面の向きがわからなくなってしまうこともあった。

それらを解決するために図6.1で示すように表示されている平面の外側に軌跡を描いた場合に、必要な分だけ平面を延長させる手法が考えられる。経路の指定と同時に平面を延長するため、旋回操作との誤作動を防ぎ、それぞれの動作を独立して行う必要がない分、スムーズに操作を行うことができる。

6.2.5 追加評価実験構想

本論文では技術的な面から行うことができなかった、想定している評価実験と実験を行う際の課題について検討する。

障害物回避操作

飛行ロボットを自由に操作する環境として、障害物のない広い空間が望ましいが、UAVのアプリケーションでもある人が侵入し難い場所の探索を行う際、障害物が多々存在する環境での操作が考えられる。そこで、障害物を回避する場面を想定した評価実験が望ましい。障害物回避例として、ロボット進行上に障害物がある際、大きさによっては迂回して

避ける動作が考えられる。提案手法では仮想平面上に障害物を迂回するための曲線軌跡を描くことで回避行動を行うことが可能であると考えられる。しかし、6.2.3節に述べた通り、現在の実装では曲率の大きい曲線移動は、空中を移動する際に生じる慣性より、記述通りの移動が困難な場合がある。そのため、現状では本評価実験を行うことは難しい。

UAV 搭載カメラを用いた、撮影対象指定操作

UAV のアプリケーションとして、探索の際に搭載されたカメラを利用したシーンが考えられる。そこでカメラ映像を操作インタフェースに表示させ、特定の対象物を捉えるといったタスクを与えた操作性評価実験を想定している。本実験を行う課題として、操作インタフェースであるタブレット PC へ、本体に搭載されたカメラ映像と、UAV のカメラ映像を表示させる方法を考える必要があり、同時ないし片方ずつ切り替えて映像を表示させる場合が考えられる。UAV を使用する場合、搭載されたカメラを利用した探索等の応用が見込まれる。本研究においても、長距離の移動は提案手法において行い、細かい制御を別に使い分ける移動を1つの操作プランとして考えている。細かい移動制御を行う一例として、機体のカメラで捉えた対象を常に中心に写るよう制御する方式が考えられる。

6.2.6 問題点と制限事項

提案手法では解決が困難な問題について述べる。飛行ロボットの利用例として、ロボットに搭載されたカメラを用いて、対象物を追いながら撮影する操作が考えられる。この操作を実現するためには、移動を行いつつ、カメラの向きを変えるため旋回を行わなければならない。しかし提案手法では移動と旋回はそれぞれ独立した操作であり、同時に行うことは不可能である。また3次元の指示を2次元に落として操作している以上、仮想平面外の操作を行うことは不可能である。例えば上昇しながら円を描く、いわゆる螺旋状の動きはできない。それらを解決するためには現在の仮想平面を任意の曲面に変更する手法が考えられるが、曲率の指示、曲面上のルート指示を現在利用している2次元平面のタッチインタフェースで行うことが困難であり、実現は難しい。

提案手法では操作端末のカメラでロボットを捉えている必要があるため、ユーザの視界内で操作を行う必要がある。また、現状では仮想平面を出現させるために AR マーカを利用しており、マーカの認識が外れてしまう範囲ではロボットを操作することはできない。現在使用しているマーカは、ユーザから 1.8 メートル程離れてしまうと認識できないため、その距離内で操作を行う必要がある。

第7章 結論

本研究では、3次元空間を自由に移動できる無線で制御可能な小型の飛行ロボットの直感的な操作を目指し、新たに仮想平面を用いた操作手法を提案した。提案手法では、飛行ロボットの移動ルートを視覚的に提示することで、ユーザは飛行ロボットとの位置関係を考慮する必要がなく、容易な指示を可能としている。また、提案手法を実現するプロトタイプを構築し、手法の評価として飛行ロボットの一般的なコントローラであるジョイスティックコントローラとの比較実験を行った。

ロボットの前後左右方向と上下方向を同時に行う斜め移動を指示する際、ジョイスティックコントローラでは同時に複数の操作を必要とするため、習熟していないユーザは独立した操作を複数回繰り返すことによって、移動を行う。一方、提案手法においては平面上の記述によって斜め移動操作に差はなく、一直線に経路を指定し、移動を行うことが可能であり、容易に目的地に到達できる。以上より、2つの操作には移動を完遂するまでに要する時間に差が生じると考えられた。実験によってその考えは裏付けられ、空間上の斜め移動を行う際、提案手法は優位であることが客観的に示された。また、実験後のアンケートによって斜め移動が容易であることに加え、操作が直感的であること、容易であることが主観的に示された。

今後の展望として、6.2節に記したように、より操作性を向上するためのシステム改善が考えられる。また、現状では仮想平面を利用するために、ロボットに搭載したマーカを用いていることから、その認識精度に依存して、操作範囲が大きく限られてしまう。それらを解決し、提案手法に対して評価実験を重ね、特定のアプリケーションへの適用を目指したい。

謝辞

本研究の作成にあたり、指導教官の小川剛史准教授から修士課程2年間に渡って、丁寧かつ熱心なご指導を賜りました。ここに深く感謝の意を表します。また、本研究を進めるに当たり、建設的な意見、指導をしてくださった、東京大学の若原恭教授、中山雅哉准教授、関谷勇司准教授、妙中雄三助教、宮本大輔助教に心より感謝いたします。2年間の大学生活の中で、多くの知識や示唆をいただいた若原・中山・関谷・小川研究室の先輩、同期、後輩の皆様には感謝します。最後に、長い大学、大学院生活を支えてくださった母に、深く深く感謝いたします。

発表文献

国内会議 (査読あり)

1. 米澤 和也, 小川 剛史, “飛行ロボット操作性向上のための仮想平面を利用した操作手法,” VR 学研報, Vol.18, No.CS-4, pp.5-8, 2013.
2. 米澤 和也, 小川 剛史, “仮想平面を利用した飛行ロボット操作システム,” DICOMO, 2014.

国内会議 (査読なし)

1. 米澤 和也, 小川 剛史, “仮想平面を利用した飛行物体操作手法の提案,” VR 学研報, Vol.18, No.CS-3, pp.107-112, 2013.

受賞

1. 優秀発表賞：米澤 和也, “仮想平面を利用した飛行物体操作手法の提案,” 融合情報学コース輪講 ポスター発表, 2014.

参考文献

- [1] Hiroki Ogawa and Tomio Watanabe. Interrobot: a speech driven embodied interaction robot. In *Robot and Human Interactive Communication, 2000. RO-MAN 2000. Proceedings. 9th IEEE International Workshop on*, pp. 322–327. IEEE, 2000.
- [2] Dairoku Sekiguchi, Masahiko Inami, and Susumu Tachi. Robotphone: Rui for interpersonal communication. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pp. 277–278. ACM, 2001.
- [3] Susumu Tachi, Naoki Kawakami, Masahiko Inami, and Yoshitaka Zaitzu. Mutual telexistence system using retro-reflective projection technology. *International Journal of Humanoid Robotics*, Vol. 1, No. 01, pp. 45–64, 2004.
- [4] Francois Michaud, Patrick Boissy, Daniel Labonte, Helene Corriveau, Andrew Grant, Michel Lauria, Richard Cloutier, Marc-André Roux, Daniel Iannuzzi, and Marie-Pier Royer. Telepresence robot for home care assistance. In *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, pp. 50–55, 2007.
- [5] Luca Tonin, Tom Carlson, Robert Leeb, and JR del Millan. Brain-controlled telepresence robot by motor-disabled people. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pp. 4227–4230. IEEE, 2011.
- [6] Fumihide Tanaka, Toshimitsu Takahashi, Shizuko Matsuzoe, Nao Tazawa, and Masahiko Morita. Telepresence robot helps children in communicating with teachers who speak a different language. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 399–406. ACM, 2014.

- [7] Alexandre Alapetite, John Paulin Hansen, and I Scott MacKenzie. Demo of gaze controlled flying. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pp. 773–774. ACM, 2012.
- [8] Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. *Journal of neural engineering*, Vol. 10, No. 4, p. 046003, 2013.
- [9] 吉田成朗, 鳴海拓志, 橋本直, 谷川智洋, 稲見昌彦, 五十嵐健夫, 廣瀬通孝. ジェスチャ操作型飛行ロボットによる身体性の拡張. *インタラクション*, pp. 403–408, 2012.
- [10] Charith Lasantha Fernando, Masahiro Furukawa, Tadatoshi Kurogi, Sho Kamuro, Katsunari Sato, Kouta Minamizawa, and Susumu Tachi. Design of telesar v for transferring bodily consciousness in telexistence. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5112–5118. IEEE, 2012.
- [11] Keita Higuchi and Jun Rekimoto. Flying head: head-synchronized unmanned aerial vehicle control for flying telepresence. In *SIGGRAPH Asia 2012 Emerging Technologies*, p. 12. ACM, 2012.
- [12] Morgan Quigley, Michael A Goodrich, and Randal W Beard. Semi-autonomous human-uav interfaces for fixed-wing mini-uavs. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Vol. 3, pp. 2457–2462. IEEE, 2004.
- [13] Masayuki Tani, Kimiya Yamaashi, Koichiro Tanikoshi, Masayasu Futakawa, and Shinya Tanifuji. Object-oriented video: interaction with real-world objects through live video. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 593–598. ACM, 1992.
- [14] Thomas Seifried, Michael Haller, Stacey D Scott, Florian Perteneder, Christian Rendl, Daisuke Sakamoto, and Masahiko Inami. Cristal: a collaborative home media and device controller based on a multi-touch display. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 33–40. ACM, 2009.

- [15] Cheng Guo, James Everett Young, and Ehud Sharlin. Touch and toys: new techniques for interaction with a remote group of robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 491–500. ACM, 2009.
- [16] Sunao Hashimoto, Akihiko Ishida, Masahiko Inami, and Takeo Igarashi. Touchme: An augmented reality based remote robot manipulation. In *The 21st International Conference on Artificial Reality and Telexistence, Proceedings of ICAT2011*, 2011.
- [17] Jun Kato, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. Multi-touch interface for controlling multiple mobile robots. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pp. 3443–3448. ACM, 2009.
- [18] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pp. 85–94. IEEE, 1999.
- [19] Shunichi Kasahara, Ryuma Niiyama, Valentin Heun, and Hiroshi Ishii. extouch: spatially-aware embodied manipulation of actuated objects mediated by augmented reality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pp. 223–228. ACM, 2013.
- [20] Daisuke Sakamoto, Koichiro Honda, Masahiko Inami, and Takeo Igarashi. Sketch and run: a stroke-based interface for home robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 197–200. ACM, 2009.
- [21] Kentaro Ishii, Shengdong Zhao, Masahiko Inami, Takeo Igarashi, and Michita Imai. Designing laser gesture interface for robot control. In *Human-Computer Interaction—INTERACT 2009*, pp. 479–492. Springer, 2009.
- [22] Hitoshi Miyanaka, Norihiko Wada, Tetsushi Kamegawa, Noritaka Sato, Shingo Tsukui, Hiroki Igarashi, and Fumitoshi Matsuno. Development of an unit type robot” kohga2” with stuck avoidance ability. In *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3877–3882. IEEE, 2007.

- [23] Andreas Wendel, Michael Maurer, Gottfried Graber, Thomas Pock, and Horst Bischof. Dense reconstruction on-the-fly. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1450–1457. IEEE, 2012.
- [24] Danielle Cummings, Stephane Fymat, and Tracy Hammond. Sketch-based interface for interaction with unmanned air vehicles. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pp. 1511–1516. ACM, 2012.
- [25] Lanah Evers, Ana Isabel Barros, Herman Monsuur, and Albert Wagelmans. Online stochastic uav mission planning with time windows and time-sensitive targets. *European Journal of Operational Research*, Vol. 238, No. 1, pp. 348–362, 2014.
- [26] Bruno S Façal, Fausto G Costa, Gustavo Pessin, Jó Ueyama, Heitor Freitas, Alexandre Colombo, Pedro H Fini, Leandro Villas, Fernando S Osório, Patrícia A Vargas, et al. The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. *Journal of Systems Architecture*, Vol. 60, No. 4, pp. 393–404, 2014.
- [27] 間野直哉, 鈴木太郎, 天野嘉春, 橋詰匠, 瀧口純一. 1a2-c14 災害情報収集のための小型 uav 搭載シンバルカメラを用いた物体追跡に関する研究. *ロボティクス・メカトロニクス講演会講演概要集*, Vol. 2010, , 2010.
- [28] Arko Lucieer, Darren Turner, Diana H King, and Sharon A Robinson. Using an unmanned aerial vehicle (uav) to capture micro-topography of antarctic moss beds. *International Journal of Applied Earth Observation and Geoinformation*, Vol. 27, pp. 53–62, 2014.
- [29] Keita Higuchi, Tetsuro Shimada, and Jun Rekimoto. Flying sports assistant: external visual imagery representation for sports training. In *Proceedings of the 2nd Augmented Human International Conference*, p. 7. ACM, 2011.
- [30] Keita Higuchi, Yoshio Ishiguro, and Jun Rekimoto. Flying eyes: free-space content creation using autonomous aerial vehicles. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 561–570. ACM, 2011.
- [31] Dan He, Haoyi Ren, Weidong Hua, Gang Pan, Shijian Li, and Zhaohui Wu. Flyingbuddy: augment human mobility and perceptibility. In *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 615–616. ACM, 2011.

- [32] Eberhard Graether and Florian Mueller. Joggerobot: a flying robot as jogging companion. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pp. 1063–1066. ACM, 2012.
- [33] Megha Sharma, James Everett Young, and Rasit Eskicioglu. Developing guidelines for in-the-field control of a team of robots. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pp. 233–234. IEEE, 2012.
- [34] Megha Sharma, Dale Hildebrandt, Gem Newman, James Everett Young, and Rasit Eskicioglu. Communicating affect via flight path exploring use of the laban effort system for designing affective locomotion paths. In *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pp. 293–300. IEEE, 2013.
- [35] 小野澤清人, 芝星帆, 吉永努, 入江英嗣. タッチ支持によるお供ロボットナビゲーション. マルチメディア、分散、協調とモバイル (DICOMO), pp. 1897–1904, 2013.
- [36] Kevin Pfeil, Seng Lee Koh, and Joseph LaViola. Exploring 3d gesture metaphors for interaction with unmanned aerial vehicles. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 257–266. ACM, 2013.
- [37] Naoji Shiroma, Noritaka Sato, Yu-huan Chiu, and Fumitoshi Matsuno. Study on effective camera images for mobile robot teleoperation. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, pp. 107–112. IEEE, 2004.
- [38] Koki Toda, Naoya Maeda, Yuta Sugiura, Masahiko Inami, and Maki Sugimoto. 繫留浮遊体を用いた三人称視点からのロボットのリアルタイム操作支援インタフェースの提案.
- [39] Tsubasa Yamamoto, Yuta Sugiura, Suzanne Low, Koki Toda, Kouta Minamizawa, Maki Sugimoto, and Masahiko Inami. Pukapucam: Enhance travel logging experience through third-person view camera attached to balloons. In *Advances in Computer Entertainment*, pp. 428–439. Springer, 2013.
- [40] 村瀬洋ほか. 2次元照合による3次元物体認識—パラメトリック固有空間法—. 電子情報通信学会論文誌 D, Vol. 77, No. 11, pp. 2179–2187, 1994.

- [41] 関晃仁, ウッドフォードオリバー. 単眼カメラの動画像を用いたリアルタイムで緻密な3次元再構成技術. 東芝レビュー= Toshiba review, Vol. 68, No. 5, pp. 40-43, 2013.