

手でカメラ操作し足で操縦するドローンUI

笠原 暢仁¹ 宮下 芳明¹

概要: ドローンは、搭載デバイスであるカメラやロボットアームによって、空撮をしたり、各種作業を行ったりと、様々な用途で用いられている。しかし、ドローンの移動を行う操縦と、搭載デバイスの操作の両方を手に持ったコントローラで行うと、操縦と操作が混同してしまうことがある。そこで本研究は、ドローンの操縦と搭載デバイスの操作の混同を減らすことを目的とし、手でカメラを操作し、足でドローンの操縦を行う UI を提案する。これにより、操縦と操作の混同を減らし、安定したドローン操縦を実現しつつ、カメラ操作を両手で行うことができるようになると思われる。

1. はじめに

ドローンを使う際、ドローンの移動を行う操縦と、カメラの操作が同時に求められることがある。多くの場合、ドローン操縦とカメラ操作をどちらも手に持ったコントローラで行うが、入力が手に集中すると、操縦と操作が混同してしまう問題がある。また、ドローンによる空撮では、ドローン操縦とカメラ操作の両立が難しく、自由なカメラワークで空撮を行うのが難しい。この問題を解決するために、ドローン操縦とカメラ操作を2名のオペレータで分担して行う製品があり、空撮で使用されている [1, 2]。しかし、2名のオペレータが高度に連携する必要がある場合において、作業を行うのは容易ではない。よって、1名のオペレータがドローン操縦とカメラ操作を混同せずに、自由なカメラワークで空撮を行えるドローン UI が望ましい。

本研究の目的は、1名のオペレータがドローン操縦とカメラ操作を混同せずに自由なカメラワークで空撮を行えるようにすることである。そこで手と足を使い、操縦と操作をそれぞれ別のインタフェースで行う手法を提案する。本稿では、足でドローンの操縦を行い、手でカメラの操作を行うシステムを考える。システムのプロトタイプとして、仮想空間において、足を伸ばす力によってドローン操縦し、手に持ったカメラ型デバイスでカメラ操作するインタフェースを試作した (図1)。

プロトタイプを用いたユーザスタディの結果、足によるドローン操縦と手によるカメラ操作を混同せず、思い通りに行えることが明らかになった。また、参加者がドローン操縦に10分程度で慣れる様子が観察されたことから、直感性の高い UI である可能性が示された。



図1: 提案システムを使用している様子。足を伸ばす力によってドローンの操縦を行い、手に持ったカメラ型のデバイスによってカメラを操作する。本稿ではプロトタイプとして仮想空間内のドローンの制御に用いる。

2. 関連研究

2.1 足を用いた入力インタフェース

増野らは、足の裏の重心変化によって仮想空間内の移動操作を行う入力インタフェース (Waraji) を開発し、Joy Stick と同等の操作性があることを明らかにした [3]。Waraji は、靴型デバイスに取り付けた感圧センサによって両足の裏の重心変化をそれぞれ取得し、仮想空間におけるキャラクタの平面移動の操作に用いる入力インタフェースである。

¹ 明治大学

Iwata らは、仮想空間での移動を足で行うためのインタフェース (Torus Treadmill) を開発し、Joy Stick や Motion Base に対して、平均総誤差距離が有意に小さくなることを明らかにした [4]。Torus Treadmill は、ユーザが移動した方向、距離に応じて足元のトレッドミルが動き、ユーザを物理世界に定位したまま、無限に歩行できるインタフェースである。

Bektaş らは、両足の座面が独立して動かせる椅子 (Limbic Chair) に座った状態で、両足の動きを入力とするインタフェースを開発した [5]。両足の動きと仮想空間内のキャラクターの移動とのマッピングについて、最も適するものを明らかにするためのユーザスタディを行った。その結果、足を閉じる動作が、仮想空間内のキャラクターの前進操作にマッピングできる可能性を示した。

2.2 ドローンの操縦インタフェース

Higuchi らは、ドローンを手に持ったコントローラで操縦するより、頭の動きと傾きで操縦する方が容易であることを明らかにした [6]。頭でドローンを制御するインタフェース (Flying Head) を開発した。Flying Head では頭の上下の位置が上下移動に、前後左右の傾きがそれぞれの方向への移動に、向きが回転操作にマッピングされた。

天間らは、2 台のドローンを空間的に連動させ、三人称視点を提供することで、ユーザがドローンの周辺状況を理解しやすくなることを明らかにした [7]。提案システムは、親ドローンと子ドローンを連動させ、親ドローンのカメラによって子ドローンの三人称視点を提供した。

2.3 仮想空間における飛行用の入力インタフェース

Zhang らは、ユーザが仮想空間においてキャラクターをヨー回転させたい時に、旋回したい方向と反対の足を押し下げる動作を行うことを、発見した [8]。これにより、回転方向と反対の足に力を入れる入力が、ヨー回転にマッピングできる可能性が示された。

Chen らは、HMD の位置と傾きによる 6DoF の移動と回転制御が、Joy Stick による制御より優れていることを明らかにした [9]。評価項目は、タスク完了時間、3次元パスの総距離、総角度、平均移動速度、平均角速度の 5 項目であり、全ての項目で提案手法が優れていた。

3. 提案システム

本稿の提案システムは、足を伸ばす力によってドローン操縦し、手に持ったカメラ型のデバイスによってカメラ操作する入力インタフェースシステムである。システムのプロトタイプとして、Unity で生成する仮想空間内のドローンとカメラを提案システムによって操縦、操作するシステムを試作した。



図 2: 足によるドローン操縦システム。両足の爪先側と踵側の紐にかかる張力を力覚センサで取得し、そのバランスによってドローンを操縦する。

3.1 足によるドローン操縦システム

3.1.1 概要

足によるドローン操縦システムの装置は、Arduino、力覚センサ、紐、板、椅子で構成される (図 2)。椅子に力覚センサを 4 つ取り付け、それぞれの力覚センサに 1 本ずつ紐を取り付ける。それぞれの紐の先端に、両足を乗せるための板の爪先側か踵側のいずれかを取り付ける。ユーザが足を伸ばした時の紐にかかる張力を力覚センサが計測し、その値を入力として仮想ドローンの移動と回転を制御する。

3.1.2 キャリブレーション

ユーザごとの力の差や、利き足と非利き足の力の差を考慮し、キャリブレーションを行う。爪先側の紐にかかる力の最大値、踵側の紐にかかる力の最大値、ユーザが自然に両足を伸ばしている状態の値を測定する。これにより、ユーザごとに力覚センサが取得する最大値を決定することで、力の差を考慮する。また、ユーザが自然に両足を伸ばしている状態を、ドローンのホバリング状態とする。

$$F_n = \begin{cases} \frac{F - F_c}{F_c} & (F < F_c) \\ \frac{F - F_c}{F_m - F_c} & (F \geq F_c) \end{cases}$$

F_n : 正規化した張力

F : 現在の張力

F_c : 適度に力を入れた状態で保存した張力

F_m : 保存した張力の最大値

上記の計算を、全てのセンサで得られた値に対して行うことによって、値を正規化する。これにより正規化した張力 F_n は、足を伸ばしていない時に -1、自然に足を伸ばしている時に 0、ユーザごとの最大値と同じ力で足を伸ばしている時に 1 となる。

3.1.3 マッピング

以下に、プロトタイプに対する入力と仮想ドローン操縦とのマッピングについて示す。

前進：両足の爪先による張力の和

後進：両足の踵による張力の和

右移動：左足の爪先と踵による張力の和

左移動：右足の爪先と踵による張力の和

上昇：両足の爪先と踵による張力の和

下降：両足の爪先と踵による張力の和

右回転：左足の爪先と右足の踵による張力の和

左回転：右足の爪先と左足の踵による張力の和

上昇と下降について、両足の爪先と踵にかかる張力がプラスの時に上昇し、マイナスの時に下降する。上記のマッピングをデフォルト設定とし、右回転と左回転のマッピングが反転した設定を反転設定とする。ユーザは、好みに合わせてどちらかを選択する。

3.1.4 移動速度・回転速度の計算

以下に、仮想ドローンの移動速度と回転速度の計算手法について示す。ただし、ユーザによるマッピングの反転を行わず、デフォルトの場合における計算とする。

仮想ドローンの移動速度ベクトル、右方向へのヨー回転速度を、下記の式によって求める。

$$\vec{V} = \vec{V}_0 + \vec{A}_f + \vec{A}_r + \vec{A}_u + \vec{Res}$$

$$V_{yawr} = V_{yawr0} + A_{yawr} + Res_{yawr}$$

\vec{V} ：移動速度ベクトル

\vec{V}_0 ：1フレーム前の移動速度ベクトル

$\vec{A}_f, \vec{A}_r, \vec{A}_u$ ：前、右、上方向への加速度ベクトル

\vec{Res} ：移動に対する抵抗ベクトル

V_{yawr} ：右方向へのヨー回転速度

V_{yawr0} ：1フレーム前の右方向へのヨー回転速度

A_{yawr} ：右方向へのヨー回転加速度

Res_{yawr} ：回転に対する抵抗

移動速度ベクトルは、1フレーム前の値に、前、右、上方向（以下3方向）の加速度ベクトルと移動に対する抵抗ベクトルを加算することによって求める。右方向へのヨー回転速度は、1フレーム前の値に、右方向への角加速度と、回転に対する抵抗を加算することによって求める。

移動に対する抵抗ベクトル、3方向の加速度ベクトル、回転に対する抵抗、右方向へのヨー回転加速度を、下記の式によって求める。

$$\vec{Res} = -\vec{V} \times P_{res}$$

$$\vec{A}_f = \vec{f} \times I_f \times P_m$$

$$\vec{A}_r = \vec{r} \times I_r \times P_m$$

$$\vec{A}_u = \vec{u} \times I_u \times P_m$$

$$Res_{yawr} = -V_{yawr} \times P_{resyawr}$$

$$A_{yawr} = I_{yawr} \times P_{yawr}$$

P_{res} ：移動に対する抵抗パラメータ

$\vec{f}, \vec{r}, \vec{u}$ ：前、右、上方向への単位ベクトル

I_f, I_r, I_u ：前、右、上方向への入力

P_m ：移動速度パラメータ

$P_{resyawr}$ ：回転に対する抵抗パラメータ

I_{yawr} ：右回転への入力

P_{yawr} ：回転速度パラメータ

それぞれの抵抗は、現在の移動方向と回転方向の反対方向に生じる加速度であり、重力、空気抵抗、ドローン制御用の抵抗を内包する。3方向の加速度ベクトルは、単位ベクトルにユーザによる入力と移動速度パラメータを乗算することで求め、右方向への角加速度は、ユーザによる右回転への入力に、回転速度パラメータを乗算することで求める。

3方向への入力と右回転への入力を、下記の式によって正規化した張力から求める。

$$I_f = \frac{F_{rtn} + F_{ltn} - F_{rhn} - F_{Lhn}}{2}$$

$$I_r = \frac{-F_{rtn} + F_{ltn} - F_{rhn} + F_{Lhn}}{2}$$

$$I_u = \frac{F_{rtn} + F_{ltn} + F_{rhn} + F_{Lhn}}{4}$$

$$I_{yawr} = \frac{-F_{rtn} + F_{ltn} + F_{rhn} - F_{Lhn}}{2}$$

F_{rtn} ：右足の爪先による張力（正規化後）

F_{ltn} ：左足の爪先による張力（正規化後）

F_{rhn} ：右足の踵による張力（正規化後）

F_{Lhn} ：左足の踵による張力（正規化後）

それぞれの入力は、両足の爪先と踵による張力の組み合わせによって求め、マイナスの値になった時には、反対方向への入力となる。



図 3: 手によるカメラ操作システム。カメラ型デバイスの向きによってドローンのカメラの向きを制御し、レンズを回転させることでズーム倍率を制御する。HMD (Meta Quest 2) にはドローンに搭載された全天球カメラの映像を映し出す。

3.2 手によるカメラ操作システム

3.2.1 概要

手によるカメラ操作システムの装置は、HMD (Meta Quest 2)、3D プリントしたカメラ型ケース、Meta Quest 2 用 Touch コントローラ (右手用)、任天堂 Switch 用 Joy-Con (左右 1 組) によって構成される (図 3)。Touch コントローラと左手用の Joy-Con をカメラ型ケースのボディ部分に取り付け、右手用の Joy-Con をレンズ部分の内部に取り付ける。Touch コントローラの向きを取得し、仮想カメラの向きを制御する。また、Joy-Con に搭載されたジャイロセンサによってレンズ部分の回転速度を取得する。これにより、仮想カメラのズーム倍率を制御する。HMD に仮想ドローンの位置に同期した全天球カメラの映像を映し出す。仮想カメラのディスプレイとして、平面状のオブジェクトに仮想カメラの映像を投影し、位置と向きを Touch コントローラに同期させた。

3.2.2 操作方法

カメラ型デバイスの向きにより、ドローンに搭載されるカメラの向きを制御する。また、レンズ部分をユーザから見て時計回りに回転させることでズームインし、反時計回りに回すことでズームアウトする。

4. ユーザスタディ

提案システムのプロトタイプによって、ドローン操縦とカメラ操作を混同せずに行えるかを検討するためにユーザスタディを行った。ユーザスタディでは、Unity で生成した仮想空間内のドローンとカメラを制御した。

4.1 機材

PC、HMD (Meta Quest 2)、提案システムのプロトタイプを使用した。HMD のリフレッシュレートは 72Hz とし、Unity のプログラムの上限 FPS は 90FPS とした。

4.2 参加者

参加者は、情報系の大学生 6 名 (男性 3 名、女性 3 名、平均年齢 21.2 歳、標準偏差 1.17 歳) であった。性差による力の違いや VR 酔いの違いを考慮するために、性別を回答させた。6 名中 3 名がドローンまたはドローンシミュレータの操縦経験があり、全員が HMD の使用経験があった。

4.3 評価

プロトタイプの制御可能性に関するアンケートと、VR 酔いに関するアンケートを実施した。制御可能性に関するアンケートは、ドローンの操縦、カメラの操作、操縦と操作のミスに関する 6 つの質問によって構成した (-3 から +3 の 7 段階リッカート尺度)。VR 酔いに関するアンケートには、The Simulator Sickness Questionnaire (SSQ) を使用し、酔いの変化を確認するために、タスクの前後で参加者に回答させ、比較した [10]。

4.4 タスク

プロトタイプによって、仮想ドローンを操縦しながらカメラを操作できるかを検討するためのタスクを作成した。仮想空間に、撮影対象となるドラゴンを配置した (図 4)。ドラゴンは、250 × 250 × 250 ユニット (1 ユニット = Unity の 1m) の範囲内から目標地点をランダムに決定し、目標地点に向かってまっすぐ飛行した。目標地点に到達後、参加者が追いつけなくなることを防ぐために、2 秒間から 3 秒間 (ランダム) その場に静止した。その後、次の目標地点を範囲内からランダムに決定した。参加者は、以上のことを繰り返し続けるドラゴンを 10 分間撮影した。撮影をする時に、参加者はできる限りカメラの画角の中心にドラゴンを捉えるように求められた。



図 4: 撮影対象となるドラゴン。参加者はできる限りカメラの画角の中心に、撮影対象を捉えるようにして、10 分間撮影した。

4.5 手順

参加者は、提案システムのプロトタイプの椅子に座り、HMDを装着してタスクを行った。タスク開始前に参加者は、足を自然に伸ばした状態で足に板を装着し、足を伸ばすことで紐に力をかけることができる状態にした。次に、SSQに回答した。回答後、ドローンの操縦方法とカメラの操作方法の説明を受けた。その後、足によるドローン操縦インタフェースのキャリブレーションを行ってから、十分な練習を行った。練習中に、左右回転の入力について、デフォルト設定と反転設定の両方を体験し、どちらにするかを選んだ。タスク終了後、SSQに回答し、続けてプロトタイプの制御可能性に関するアンケートに回答した。

5. 結果

アンケートにおいて、参加者全員が提案システムのプロトタイプの操作性を高く評価した(図5)。VR酔いにおける悪心の項目で酔いが増加しなかった(図6)。図のエラーバーは標準偏差を示している。

5.1 ドローンの操縦

提案システムのプロトタイプによるドローンの操縦は、十分な操作性があることが示唆された。ドローンの操縦の回転方向を反転させた参加者は6名中3名であった。ドローンの操縦に関するアンケートの結果(図5-A)、ドローンの操縦のしやすさの項目(I)では、回答の平均値が+1.83となった。思い通りにドローンを操縦できるかの項目(II)では、回答の平均値が+1.83となった。

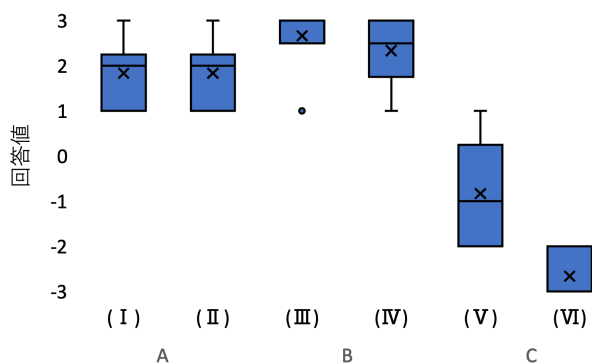


図 5: A: ドローンの操縦に関するアンケート結果
(I) ドローン操縦のしやすさ (II) 思い通りにドローン操縦できるか
B: カメラの操作に関するアンケート結果
(III) カメラ操作のしやすさ (IV) 思い通りにカメラ操作できるか
C: 操縦と操作のミスに関するアンケート結果
(V) 操縦と操作のミスをしたか (VI) 操縦と操作の混同をしたか

参加者が10分以内に、ドローンの操縦に慣れる様子が観察された。自由記述のアンケートでは、操縦しやすい、分かりやすいといった肯定的な回答があった一方で、回転が遅い、足が疲れる、下降しながら前進できないといった否定的な回答があった。

5.2 カメラの操作

提案システムのプロトタイプによるカメラの操作は、高い操作性があることが示唆された。カメラの操作に関するアンケートの結果(図5-B)、カメラの操作のしやすさの項目(III)では、回答の平均値が+2.67となった、思い通りにカメラ操作ができるかの項目(IV)では、回答の平均値+2.33となった。

自由記述のアンケートでは、カメラの操作に集中できる、分かりやすいといった肯定的な回答があった一方で、レンズの回転方向を間違えることが多かった、ズームが速すぎて細かく調整できなかったといった否定的な回答があった。

5.3 操作ミス

提案システムのプロトタイプによるドローンの操縦とカメラの操作は、操縦と操作を混同しないことが明らかになった。操作ミスに関するアンケートの結果(図5-C)、操縦と操作のミスをしたかの項目(V)では、回答の平均値が-0.83となった。操縦と操作の混同をしたかの項目(VI)では、回答の平均値が-2.67となった。

5.4 VR酔い

提案システムのプロトタイプによるドローンの操縦とカメラの操作は、VR酔いを引き起こしにくいことが示唆された。タスク前後のSSQスコアを比較したところ、悪心(Nausea)の項目においてスコアが増加せず、総合スコア(Total Severity)、失見当識(Disorientation)、眼精疲労(Oculomotor)の項目においてスコアが増加した(図6)。

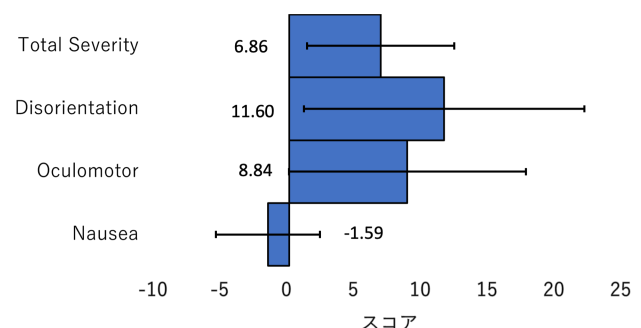


図 6: The Simulator Sickness Questionnaire (SSQ) の結果。総合スコア(Total Severity)、失見当識(Disorientation)、眼精疲労(Oculomotor)の項目では酔いが増加したが、悪心(Nausea)の項目では酔いが増加しなかった。

6. 考察

プロトタイプによるユーザスタディの結果から、提案システムによって、ドローン操縦とカメラ操作を混同せず、思い通りに行えることが明らかになった。また、VR酔いにおける悪心の項目を増加させない可能性が示された。

6.1 ドローン操縦とカメラ操作

参加者はドローン操縦とカメラ操作の両方について、しやすい、思い通りに動かせるの2つの指標で提案システムを高く評価した。また、参加者は練習を始めてから、10分以内にドローンの操縦に慣れることができた。これらのことから、提案システムは操作性と直感性の高いインタフェースである可能性が示された。また、ドローン操縦とカメラ操作の混同についての回答値が低いことから、ユーザはドローン操縦とカメラ操作を混同しにくいことが明らかになった。これは、手と足にインタフェースを分けたことが影響していると考えられる。

自由記述のアンケートにカメラ操作に集中できる趣旨の回答があった。このことから、提案システムを用いることで、カメラの操作をより細かく行える可能性が示された。これにより、これまでのドローンでは難しかった自由なカメラワークでの空撮ができると考えられる。

足によるドローン操縦インタフェースが高い操作性を持っている可能性が示された。このことから、足でドローンを操縦しながら、手によってロボットアームを操作し、様々な道具を遠隔で使用するという応用も考えられる。

6.2 VR酔い

SSQの悪心の項目において、スコアが増加しなかったことから、酔いを増加させないインタフェースである可能性が示された。これは、仮想空間に表示したカメラのディスプレイが注視点として機能し、酔いを軽減した可能性がある。また、タスクに使用した仮想空間がオブジェクトの少ない環境だったことで、オプティカルフローが小さかったことも影響した可能性がある。一方で、悪心以外の項目ではスコアが増加していることから、酔いを無くす事はできないことが明らかになった。

7. 制約・展望

本稿では、仮想空間におけるタスクしか行っていないため、実際のドローンに本稿の知見を利用できるかは明らかでない。今後は実際のドローンを用いて提案システムの有用性を検討する予定である。また、提案システムと既存システムの比較実験を行っていないため、提案システムの有用性は定量評価されていない。今後は操作性、直感性に関する定量評価と比較実験を行う予定である。

8. おわりに

本稿では、足を伸ばす力によってドローンを操縦し、手に持ったカメラ型のデバイスでカメラを操作するシステムのプロトタイプを試作した。プロトタイプを用いたユーザスタディを行った結果、ドローン操縦とカメラ操作を混同せず、思い通りに行えることが明らかになった。また、参加者がドローン操縦に10分程度で慣れる様子が観察されたことから、提案システムが直感的で操作性の高いインタフェースである可能性が示された。これらのことから、提案システムを用いることで実際のドローンでも、1名のオペレータによってドローン操縦とカメラ操作を混同せず、思い通りに行うことができると考える。また、カメラ操作に集中できることから、通常のドローンでは難しい自由なカメラワークで空撮を行えるようになると思う。

参考文献

- [1] CIRCLE 空撮課: 空撮課 Information05, CIRCLE 空撮課 (オンライン), 入手先 (<https://drone.circle-j.co.jp/info/05/>) (参照 2022-07-05).
- [2] DJI: Inspire2, DJI (online), available from (<https://www.dji.com/jp/inspire-2>) (accessed 2022-07-05).
- [3] 増野智経, 齋藤豪, 高橋裕樹, 中嶋正之: Waraji: 仮想環境のための足を用了入力インタフェース, 映像情報メディア学会誌, Vol. 54, No. 6, pp. 833-839 (2000).
- [4] Hiroo, I.: The torus treadmill: Realizing locomotion in VEs, *IEEE Computer Graphics and Application*, Vol. 19, No. 6, pp. 30-35 (1999).
- [5] Bektaş, K., Thrash, T., van Raaij, M. A., Künzler, P. and Hahnloser, R.: The systematic evaluation of an embodied control interface for virtual reality, *Plos one*, Vol. 16, No. 12, p. e0259977 (2021).
- [6] Keita, H., Katsuya, F. and Jun, R.: Flying head: A head-synchronization mechanism for flying telepresence, *International Conference on Artificial Reality and Telexistence (ICAT)*, Vol. 23, pp. 28-34 (2013).
- [7] 天間遼太郎, 高嶋和毅, 藤田和之, 末田航, 北村喜文: 空間連動する2つのカメラ視点を用了ドローン操縦インタフェースの拡張, 情報処理学会論文誌, Vol. 61, No. 8, pp. 1319-1332 (2020).
- [8] Zhang, Y., Riecke, B. E., Schiphorst, T. and Neustaedter, C.: Perch to fly: Embodied virtual reality flying locomotion with a flexible perching stance, *Designing Interactive Systems Conference*, pp. 253-264 (2019).
- [9] Chen, W., Plancoulaine, A., Férey, N., Touraine, D., Nelson, J. and Bourdot, P.: 6DoF navigation in virtual worlds: comparison of joystick-based and head-controlled paradigms, *19th ACM Symposium on Virtual Reality Software and Technology*, pp. 111-114 (2013).
- [10] Kennedy, R. S., Lane, N. E., Berbaum, K. S. and Lilienthal, M. G.: Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness, *The international journal of aviation psychology*, Vol. 3, No. 3, pp. 203-220 (1993).