

Reliabrella: 周囲環境により自動で最適な方向を向く ハンズフリー傘

土屋 慶吾¹ 青山 泰志¹ 成尾 怜真¹ 中山 裕貴² 濱川 礼¹

概要: 本論文では, 風速・風向をはじめとした雨の角度に関わる周囲環境データを基に, 背中に背負った傘を自動で最適な角度に調節するシステム「Reliabrella」について述べる. 今日ではハンズフリーを実現した傘が販売されているものの, 実際に使用している者はほとんど見られない. その原因として, ユーザ自身が傘を持つ場合と異なり, 風の変化によって傘の角度を任意に変えられないことから, 防雨性能が従来の傘と比較して著しく劣ることなどがあると考えた. そこで本研究では, ハンズフリー傘を周囲環境により自動で最適な角度に調節するシステム「Reliabrella」を開発した. Reliabrella は, 操作の iOS アプリによって身体, 荷物に対して雨から守る優先度を付けることが可能で, その優先度に合わせて傘を雨に対して適切な角度に調整する. そのため, 従来のハンズフリー傘以上の防雨性能がある.

Reliabrella: A Hands-free Umbrella that Faces the Optimal Direction Automatically by the Surrounding Environment

KEIGO TSUCHIYA¹ TAISHI AOYAMA¹ RYOMA NARUO¹ YUKI NAKAYAMA²
REI HAMAKAWA¹

1. 背景

傘の歴史は古く, 4000 年以上前から存在していると言われている. 元来, 傘は日傘として使用されていた. 13 世紀に入り, 初めて開閉式の傘が生まれ, 18 世紀後半になると現在の傘とほぼ同じ構造の雨傘が普及した. また, 日本での洋傘の最古の記録は, 1804 年に長崎に入港した中国船の舶載品目であると言われている.

傘には大きく分けて洋傘と和傘の 2 種類が存在する. 洋傘は用途別に雨傘, 晴雨兼用雨傘 (雨晴兼用傘), 晴雨兼用日傘 (晴雨兼用パラソル), 日傘 (純パラソル) の 4 種類に分類される. また, 洋傘の形態別に長傘, 折傘, ショート傘, ミニ傘の 4 種類に分類され, 洋傘の形別に標準張り, 深張り, 逆張りの 3 種類に分類される [1][2][3].

近年の傘の傾向として, From プラネットが行った意識調査 [4] では, 傘選びは価格と大きさを重視すると書かれている. これは大きい傘は雨を防ぎやすく, かつなるべく安い傘を買いたいと考える人が多いためであると考えられる.

また, 欲しい傘のアンケートについて [4] では, 「軽くて丈夫な傘」「安い傘」という回答の他に「手で持たなくていい傘」という回答がある.

実際, 今日ではハンズフリーを実現した傘が多く販売されている [5][6][7][8][9]. ハンズフリー傘はハイキングや釣り, 農作業など屋外での活動を簡単かつ効率的に行うことを目的に開発された. ハンズフリー傘には「Nubrella」[10] のような背負うもの, 「アタマンプレラ」[11] のような被るもの, 「Swing handsfree」[12] のような取り付けるもの等, 様々な種類が存在する. ハンズフリー傘は便利である一方, 風により変化する雨の角度に対して傘の角度をリアルタイムに変更することは困難である. 実際, 傘の角度を任意に変更可能なハンズフリー傘として [13] が存在するが, これは傘の角度の変更時に後方にあるクランプナットで固定する必要がある. そのため, 傘の角度の変更までかなりの時間を要してしまい, 傘の角度を変更している間に雨に打たれてしまうと考えられる. また, クランプナットは一つのみであり, 一支点の駆動となるため, 吹き付ける雨に対して防雨性能が低いと考えられる. そこで, この

¹ 中京大学 工学部 情報工学科

² 中京大学 大学院 工学研究科 情報工学専攻

問題を解決するため傘のシャフト部分を三支点の駆動にし、プログラムを用いて傘の角度を制御を行うことでリアルタイムに傘を雨に対して最適な角度に調節するシステム「Reliabrella」を考案して実装した。Reliabrella では、傘のシャフト部分に2つの180°サーボモータと1つの360°サーボモータを取り付け、角度の変更を可能にしている。また Reliabrella には、風速計、風向計を始めとした後述する各種センサを搭載している。これにより、リアルタイムに変化する雨の角度に対応することが可能である。なお「Reliabrella」は、頼もしいことを意味する「Reliable」と、傘を意味する「Umbrella」から命名した。

2. 関連研究

「ハンズフリー」や「傘」に関する研究は今日まで様々なものが行われている。関連する特許と研究について以下に述べる。

2.1 Automated Hands-Free Umbrella

Lazo-Antunez らは傘を使用者の頭上で浮上させる傘の特許を取得した [14]。傘天蓋部を組み込みガスによって一定の高度で浮上するよう制御し、傘の各親骨の付け根に配置したビデオカメラから収集した画像データをもとに、使用者の頭上を一定の高度で追従することでハンズフリーを実現している。ガスによって一定の高度で浮遊しているため雨粒に対して傘の角度を調整することが出来ない点が、Reliabrella と異なる。

2.2 Eyeglass-based hands-free videophone

木村らのハンズフリーでテレビ電話を行う研究がある [15]。この研究では、携帯電話を持たずにテレビ電話を行うことのできるメガネ型テレビ電話を提案している。これはテレビ電話で顔を映す際の、顔に携帯電話をかざすという行為から発生する疲労感の解決を目的としている。Reliabrella では、ハンズフリーを実現することで傘を手を持つ際に感じられる疲労感を解決する。

2.3 Pileus Internet Umbrella

松本らの雨の日の傘による街歩きを快適にする研究がある [16]。傘上面に大型のビジュアルスクリーン、ワイヤレスインターネット接続、カメラ、モーションセンサーと位置センサーを備えており、雨の日をより楽しくすることを目的としている。Reliabrella では、ハンズフリーを実現することで両手が空き、雨の日を快適に過ごすことが出来ると考える。

3. 提案手法

Reliabrella では、風向・風速や降雨強度 (1 時間あたりの降水量 [mm/h])、ユーザの身長、ユーザが持つ荷物の位置

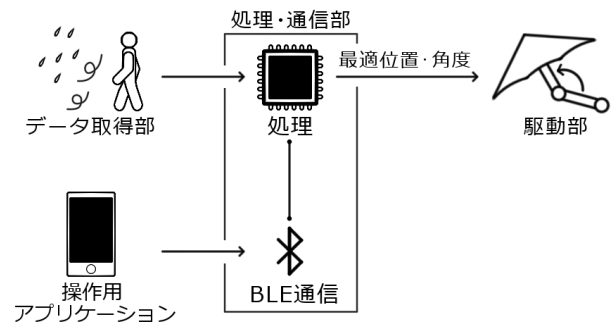


図 1: Reliabrella 構成図

を基にして、指定したユーザの身体と荷物を雨から守るために最適な傘の位置・角度を計算する。また、ユーザの歩行速度や前方の深度マップなどを取得することで簡易的な人・物体検知を実装し、すれ違う人や障害物との傘の接触を防ぐために傘の動作制限を行う。Reliabrella の構成を図 1 に示す。Reliabrella はデータ取得部、操作用アプリケーション、処理・通信部、駆動部から構成する。データ取得部では、風向・風速と歩行速度、前方の深度マップを得るために必要なデータを取得し、処理・通信部へ渡す。操作用アプリケーション (以下操作用アプリ) は処理・通信部と Bluetooth Low Energy (以下 BLE) 通信を行い、降雨強度やユーザの身長、荷物の位置の指定の他、Reliabrella の動作の変更をする際に用いる。処理・通信部では、データ取得部や操作用アプリから渡されたデータを処理し、それを基にユーザの身体や荷物を雨から防ぐために最適な傘の位置・角度を計算して駆動部を動作させる。駆動部は処理・通信部から信号を受け取り、傘の位置・角度を変更する。

3.1 デバイス概要

Reliabrella のデバイスは、データ取得部、処理・通信部、駆動部をハーネスにより背中に固定出来るウェアラブルデバイスとして設計し、デバイスの電源供給は搭載したバッテリーにより行う。駆動部は2つの180°サーボモータと1つの360°サーボモータから構成する。なお、ここでは図 2 に示されている通り、下から第1、第2、第3サーボとする。この内、第1サーボが360°サーボモータ、第2、第3サーボが180°サーボモータである。この3つのサーボで構成することにより第1サーボが向いている方向に関して、第3サーボの位置を前後に第2・第3サーボ間の長さの範囲で調整出来る。最適位置・角度の計算はこの範囲内で行う。

3.2 データ取得部

本節では、Reliabrella のデータ取得部において各データを取得する目的・手法について述べる。

3.2.1 風向・風速

風向は雨が吹き付ける向きを、風速は雨の傾斜を計算す

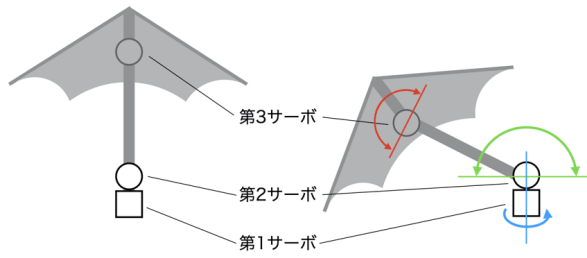


図 2: (左) サーボモータの構成と (右) 角度を変えた様子

る際に用いる。可能な限り軽量・小型化してデバイスの装着感を向上させる為、市販の風向・風速計は使用せずに自作する。風向は角度センサに風向板を取り付けることで取得する。風速は風杯型風速計と同様の回転体を製作し、風杯の回転速度が風速に比例する [17] ことを利用して、回転体の回転速度を計算して風速へと変換することで取得する。風速 1.0m/s において風速計が一定の角度を回転する際にかかる時間を $Time_{1mps}[s]$ 、計測時に風速計が同じ一定の角度を回転する際にかかった時間を $Time_{measured}[s]$ とすると、風速 1.0m/s に対して掛ける係数 $Velocity_{coef}$ は以下の式で求められる。

$$Velocity_{coef} = \frac{Time_{mps}}{Time_{measured}}$$

この係数を以下の様に風速 1.0m/s に対して掛けることで風速 $Velocity_{wind}[m/s]$ が得られる。

$$Velocity_{wind} = 1.0 \times Velocity_{coef}$$

3.2.2 ユーザ前方の深度マップ

Reliabrella の実用環境においては、人とすれ違う時などに安全上の理由でデバイスの動作を制限させる必要がある。その際に進行方向に存在する人や物体を検知するために、ユーザ前方の深度マップを取得する。深度マップは位置を左右にずらした USB カメラより得られる 2 枚の画像から生成する。

3.2.3 歩行速度

歩行速度は深度マップから進行方向の人や物体を検知した際に、それがユーザに接近するタイミングを計算するために用いる。加速度センサをデバイスに取り付け、加速度の変化により歩数をカウントし、歩幅を歩数に掛けることで移動距離を算出する。この移動距離を時間で割ることで歩行速度を取得する。

3.3 処理・通信部

処理・通信部では、データ取得部や操作用アプリから受け取ったデータの処理や、傘の最適位置・角度の計算を行い駆動部を動作させる。バッテリー搭載のウェアラブルデバイスとして設計するため、操作用アプリからのデータ受信には消費電力の低い BLE 通信を用いる。

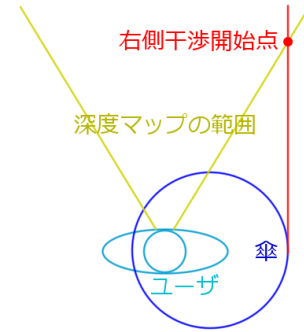


図 3: 右側衝突開始点

3.3.1 人・物体検知

Reliabrella の実用環境を想定し、人や物の近くを通った際に傘が当たることを防ぐため、簡易的な人・物体検知を行いサーボの制限角度を設定する。人・物体検知にはユーザ前方の深度マップを用いる。ここで第 1 サーボを 90° 、第 2 サーボを 0° 、第 3 サーボを 90° とした時の傘の右端の位置が右側における傘の最大稼働範囲であり、その地点から進行方向と平行に伸ばした直線と深度マップの範囲との交点を右側干渉開始点 (図 3) とする。この地点にある物体の深度を $Depth_{hit}$ とおくと、ユーザの位置を中心として分割した深度マップの右側において、 $Depth_{hit}$ 以上に近い深度で現れた物体はユーザの右側に衝突する可能性がある。Reliabrella においては人・物体検知は主題ではないため、このような深度の物体が現れた場合は歩行速度を用いてその物体を通り過ぎるまでの時間を計算し、その時間が経過するまで傘が一切右側に動作しないように第 1 サーボの角度を 180° 以上に制限する。また左側も同様にして、ユーザの左側に物体が衝突する可能性がある場合には第 1 サーボの角度を 180° 以下に制限する。

3.3.2 最適位置・角度計算

データ取得部や操作用アプリから受け取ったデータから、傘の最適位置・角度を計算する。計算においては、図 4 に示すように x 軸をユーザの右方向、 y 軸をユーザの進行方向、 z 軸をユーザの上空方向とし、原点はユーザの足元とする。更に z 軸を回転軸とした際の雨が吹き付ける角度を $\theta_{dir}[\text{rad}]$ 、降雨方向ベクトルと z 軸がなす角度を $\theta_{inc}[\text{rad}]$ とし、 $\theta_{dir}[\text{rad}]$ 方向を x' 軸とする。なお、座標の値の単位はメートルとする。

(1) ユーザに対する降雨方向を計算

3 次元における雨の角度 $\theta_{dir}[\text{rad}]$ と $\theta_{inc}[\text{rad}]$ を計算する。雨の傾斜は雨滴の落下速度と風速の合成により求められるため、まず雨滴の落下速度を計算する必要がある。求められた雨滴の落下速度と風速を合成し、そのベクトルのユーザに対する角度を求める。

(2) 身体と荷物の位置関係を計算

ユーザの身体と荷物について、 xyz 空間から $x'z$ 平面に射影した時の位置関係を計算する。計算の簡略化の

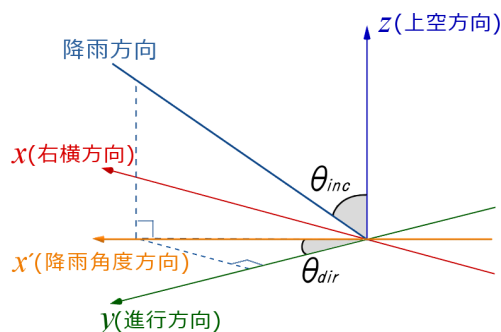


図 4: 計算における座標軸

ため、図 5 に示すようにユーザの身体と荷物はそれぞれ円柱、球体として扱う。これらの立体図形を $x'z$ 平面に射影すると、円柱と球体はそれぞれ矩形と円として扱える。

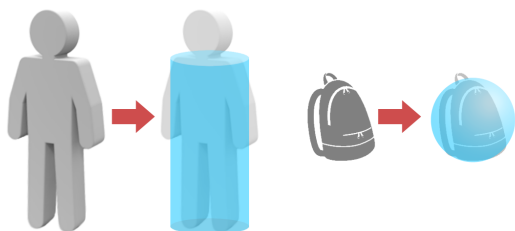


図 5: 計算における身体と荷物の扱い

(3) 最適位置・角度を計算

事前に設定した荷物の優先度を考慮し、ユーザの身体と荷物を雨から防げるような傘の最適位置・角度を $x'z$ 平面において計算する。なお、荷物の優先度は p ($0 \leq p \leq 1$) とする。この値が 1 に近いほど荷物を優先して雨から守り、逆に 0 に近いほどユーザの身体を優先して守る。図 6 に示すように、まず矩形と円の中心からそれぞれ x' 軸へ降雨方向と傾きが同じ直線を伸ばし、それぞれの x' 軸との交点を円から矩形の方向に $p-1:p$ で分割する点を求める。その点から伸ばした降雨方向と傾きが同じ直線を最適位置・角度直線 $l_{optimal}$ とする。

(4) 最適位置・角度をサーボの角度に変換

求められた最適位置・角度を実現するための各サーボの角度を計算する。第 1 サーボの角度は $\theta_{dir}[\text{rad}]$ となる。第 2 サーボの角度は第 2 サーボの稼働範囲を表す半円の中心点と、その半円と直線 $l_{optimal}$ の交点を結んだ直線が第 3 サーボの回転軸との間になす角度である。また、第 3 サーボの角度は $(\pi - \theta_{inc})[\text{rad}]$ から第 2 サーボの角度を引いた値になる。ただし、各サーボの角度によってはユーザの身体やデバイス自体に干渉する場合があるため、各サーボには角度制限を設ける必要がある。

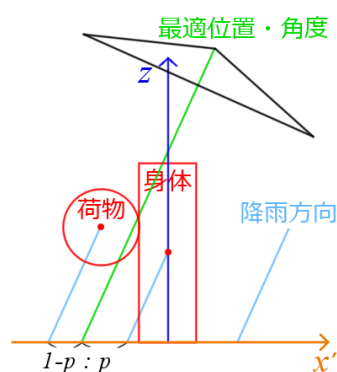


図 6: 最適位置・角度直線

(5) 角度制限を適用各サーボに対して設けられた角度制限を適用する。制限角度は、ユーザの頭部やデバイス自体に干渉しないよう事前に設定された角度と、人・物体検知によりリアルタイムで変化する角度の 2 種類がある。前者は実際に手作業で干渉角度を計測する。この 2 つの制限角度の内、より小さい角度を実際の制限角度として設定する。この制限角度を前述の計算で求められた角度が超えた場合、実際に動作させる角度は制限角度と同じ値にする。

3.4 駆動部

計算した角度を基に、実際に各サーボを動作させる。各サーボはユーザとデバイスへの負担軽減のため、速度を制限して動作させる。

3.5 操作用アプリ

操作用アプリは図 7 に示すように、降雨強度やユーザの身長、荷物の位置を処理・通信部に送信するために用いる。荷物の位置は、荷物に見なした円を視覚的に配置して指定する。送信されたデータに基づいてプログラムが実行され、最適位置・角度に傘が駆動する。

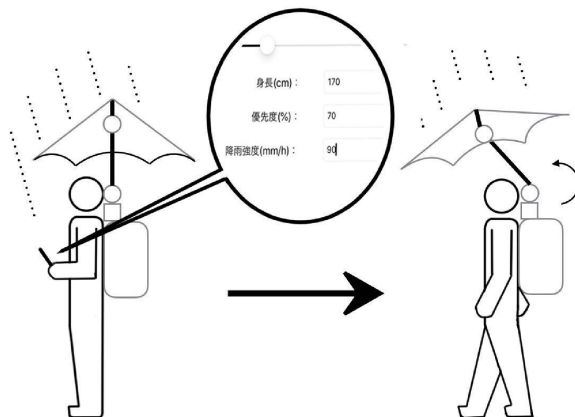


図 7: 操作用アプリの使用イメージ

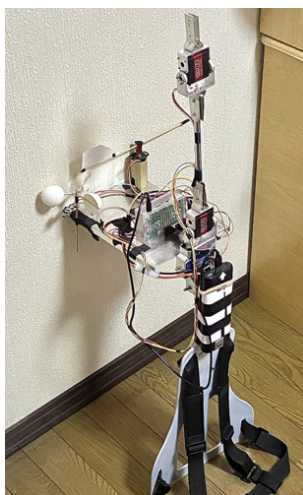


図 8: 傘を外した状態のデバイス

4. 実装

本章では、提案した手法の具体的な実装について述べる。なお、各種実装にはライブラリが豊富な Python を用いる。

4.1 デバイス製作

本節では、実際に製作したデバイスについて述べる。デバイスのフレームは軽量かつ頑丈でありながら、加工が比較的容易なアルミ板を素材に採用した。完成したデバイスの総重量は 2.1kg となった。図 8 に製作したデバイスの外見を示す。なおデバイスの主要部分を見やすくするため、傘は外した状態である。

4.1.1 デバイス構成

本項では、デバイスに使用した機器・電子部品について述べる。なお、抵抗や配線等の汎用的な部品については説明を省く。電力の供給には、第 2、第 3 サーボに対しては電圧 7.2v、容量 5000mAh のバッテリーを、その他の部品に対しては電圧 5.0v、容量 20000mAh のバッテリーを用いた。各部品の構成を図 9 に示す。

- **Raspberry Pi 3 Model B+**

処理・通信部を担う。画像処理を行うために、モデルは研究開始当時で最も処理性能が高い物を選んだ。

- **180° サーボモータ LewanSoul LD-220MG × 2**

180° の範囲で動作するサーボモータ。第 2、第 3 サーボとして使用する。ある程度の風に耐えられるよう、20kg・cm の高トルクな物を選んだ。PWM 制御により角度を指定して変化させることが出来る。

- **360° サーボモータ SPT5325LV-360**

360° 連続回転するサーボモータ。第 1 サーボとして使用する。ある程度の風に耐えられるよう、25kg・cm の高トルクな物を選んだ。PWM 制御により回転速度を変化させることが出来るが角度は指定出来ないため、別途角度を読み取る機構が必要となる。

- **サーボドライバ PCA9685**

I2C 通信により正確な PWM 信号を出力することが出来る電子回路。Raspberry Pi のみでは正確な PWM 制御は同時に 2 つまでしか行えないため、第 1 サーボの制御をこれで行う。

- **ロータリーエンコーダ RES20D-50-201-1**

AB 相 50 パルス出力のインクリメンタル形ロータリーエンコーダ。第 1 サーボの角度変化を読み取るために使用する。1 回転につき A 相・B 相からそれぞれ 50 パルスが 1/4 周期ずれて出力される。

- **フォトセンサ EE-SX674-WR 1M**

物体が発光部と受光部の間にある際に反応する光センサ。第 1 サーボの絶対角度を読み取るために使用する。

- **磁気センサ KY-003**

磁気に応答するセンサ。風杯に磁石を取り付けた風速計の回転速度を取得するために使用する。

- **角度センサ P3022-V1-CW360**

角度を電圧値として出力するセンサ。風向を取得するために使用する。ロータリーエンコーダより角度変化を読み取る精度は低いが、回転に必要なトルクが小さいため、より弱い風に対して反応出来る。

- **A/D コンバータ MCP3008**

アナログ入力された電圧値をデジタル出力する IC チップ。角度センサの電圧を読み取るために使用する。

- **加速度センサ GY-521**

3 軸の加速度センサ。歩行速度の取得において歩数をカウントする際に、歩行の振動を検出するために使用する。

- **USB カメラ Logitech HD Webcam C270n × 2**

最大解像度が 1280×720px、フレームレート 30fps の USB カメラ。人・物体検知における深度マップの生成に 2 つ使用する。

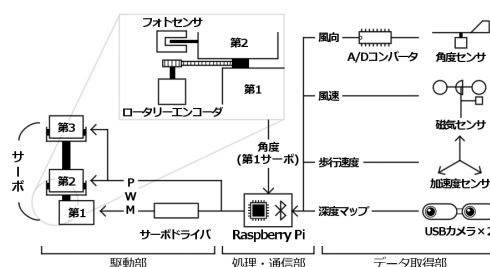


図 9: 各部品の構成

4.2 風向・風速の取得

風向は角度センサから出力される電圧によって求める。角度センサに 5v を入力した際、出力された電圧 $V_{dir}[v]$ から角度 $\theta[^\circ]$ を求める式は以下の通りになる。

$$\theta = 360 \times \frac{V_{dir}}{5}$$

一方で風速は第 3.2.1 項の式に基づいて計算する．風速計の 3 つの風杯にそれぞれ磁石を取り付け，磁気センサで風杯の通過を検出する方式であるため，風速計が 1/3 回転するごとに風速を更新する．風速 1m/s の時に風速計が 1/3 回転する際にかかる時間は，市販の風速計を使用して計測したところ 0.26s であったため，磁気センサが磁気を検出した際の前回の検出からの経過時間 $t[s]$ を用いて，風速 1.0m/s に対して掛ける係数 V_{coef} を求める式は以下の通りになる．

$$V_{coef} = \frac{0.26}{t}$$

これに風速 1m/s を掛けることで，風速を得ることが出来る．風速の更新は，新たに磁気センサが反応した瞬間と，その時点で磁気センサが反応した場合の風速が現在設定されている風速を下回る場合に行う．

4.3 歩行速度の取得

歩行速度は，加速度センサから読み取った y 軸（進行方向）と z 軸（上空方向）の加速度を合成した値の変化によって求める．これは進行方向に移動した際の加速度と，足を着地した際の振動を基に歩行を検出するためである．合成加速度が設定した上方閾値に到達した際にフラグを 0 に，下方閾値に到達した際にフラグを 1 に変更し，フラグが 1 から 0 に変化する度に歩数をカウントする．今回設定した上方閾値は -0.80m/s^2 ，下方閾値は -1.38m/s^2 である．これは 100 歩実際に歩いた際の加速度を計測したデータを基に得られた合成加速度のデータを基に設定した．上方閾値は 1 歩の最大合成加速度の中の最小値を，下方閾値は 1 歩の最小合成加速度の中の最大値を設定している．カウントした歩数に対して歩幅を掛けることで移動距離を得る．歩幅は翁長らの身長から歩幅を求める手法を用いる [18]．操作用アプリから指定したユーザの身長 $h[\text{m}]$ を用いて，歩幅 $l_{step}[\text{m}]$ は以下の式で求められる．

$$l_{step} = 0.26h + 0.31$$

得られた移動距離を時間で割ることで，移動速度を得る．移動速度の更新は，新たに 1 歩がカウントされた瞬間と，その時点で 1 歩をカウントした場合の移動速度が現在設定されている移動速度を下回る場合に行う．

4.4 人・物体検知

深度マップの生成では，2 つの USB カメラから取得した $640 \times 360\text{px}$ の画像 2 枚をステレオグラムとして使用し，OpenCV の StereoBM メソッドを用いた．なお，深度マップ生成の前には同じく OpenCV を用いたテンプレートマッチングにより画像の上下方向のズレを補正した．物体を同

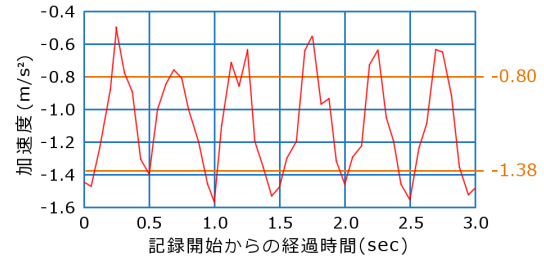


図 10: 歩行時の実際の YZ 合成加速度 (6 歩分)

色のピクセルの塊として検出するため，深度マップはピクセル値 0~7 の 8 階調に減色した．実測した所，右側干渉開始点と左側干渉開始点の物体の深度は共に 6 であったため，左右でそれぞれ 6 以上の深度が検知された場合はそれぞれの方向への一切の角度変化を行わないようにした．

4.5 最適位置・角度の計算

取得したデータを基に各種値を求め，最適位置・角度を計算する．

4.5.1 降雨角度の計算

最初に 3 次元における雨の角度を計算する．

(1) 雨滴の落下速度

雨の傾斜は雨滴の落下速度と風速の合成により求められるため，まず雨滴の落下速度を計算する必要がある．落下速度の計算には Ishwar らの降雨強度を利用した手法を用いる [19]．降雨強度を $I[\text{mm/h}]$ とすると，雨滴の落下速度 $V_r[\text{m/s}]$ は以下の式で求められる．

$$V_r = 4.5I^{0.107}$$

(2) ユーザに吹き付ける雨の角度

次に風向を $\theta_w[\text{rad}]$ ，風速を $V_w[\text{m/s}]$ とすると， y 軸（進行方向）と雨がなす角度 $\theta_{dir}[\text{rad}]$ と z 軸（上空方向）と雨がなす角度 $\theta_{inc}[\text{rad}]$ は，雨滴の落下速度 $V_r[\text{m/s}]$ を用いて以下の式で求められる．

$$\theta_{dir} = \arctan \frac{V_w \sin \theta_w}{V_w \cos \theta_w}$$

$$\theta_{inc} = \arctan \frac{V_w}{V_r}$$

4.5.2 最適位置・角度の計算

次に，ユーザの身体と荷物の位置・優先度を考慮した最適位置・角度を計算する．ここで荷物の優先度とは，操作用アプリから 0~1 の範囲で設定する小数値であり，この値が 1 に近いほど荷物を優先して雨から守り，逆に 0 に近いほどユーザの身体を優先して守る．最適位置・角度を求めるにあたって，ユーザの身体は高さ $h'[\text{m}]$ の円柱として，荷物は球体として扱う．なお， $h'[\text{m}]$ は操作用アプリから指定したユーザの身長 $h[\text{m}]$ から青年男女の平均頭長 0.19m [20] を引いた値とした． xyz 空間における中心座標が円柱は $(0, 0, h/2)$ の位置に，球体は操作用アプリから指

定した位置 (x_b, y_b, z_b) になるよう配置する.

まず円柱と球体を $x'z$ 平面にそれぞれ矩形, 円として射影する. 円柱は z 軸を中心とするため, 矩形の $x'z$ 平面における中心座標は $(0, h/2)$ となる. 一方で円の $x'z$ 平面における中心座標は z 座標は変わらず, x' 座標 $x'_b[\text{m}]$ は $\theta_{dir}[\text{rad}]$ によって場合分けされた以下の式で求められる.

$$x'_b = \begin{cases} \frac{x_b \tan \theta_{dir} + y_b}{\sqrt{1 + \tan^2 \theta_{dir}}} & (0 < \theta_{dir} < \frac{\pi}{2}) \cup (\frac{\pi}{2} < \theta_{dir} < \pi) \\ -\frac{x_b \tan \theta_{dir} + y_b}{\sqrt{1 + \tan^2 \theta_{dir}}} & (\pi < \theta_{dir} < \frac{3}{2}\pi) \cup (\frac{3}{2}\pi < \theta_{dir} < 2\pi) \\ y_b \cos \theta_{dir} & (\theta_{dir} = 0) \cup (\theta_{dir} = \pi) \\ x_b \sin \theta_{dir} & (\theta_{dir} = \frac{\pi}{2}) \cup (\theta_{dir} = \frac{3}{2}\pi) \end{cases}$$

次に求められたそれぞれの中心座標から傾きが $1/\tan \theta_{inc}$ の直線を x' 軸へ伸ばす. 矩形の中心座標から伸ばした直線と x' 軸との交点の x' 座標 $x'_{u0}[\text{m}]$ と, 同様にした円の x' 座標 $x'_{b0}[\text{m}]$ はそれぞれ以下の式で求められる.

$$\begin{aligned} x'_{u0} &= -\frac{h'}{2} \tan \theta_{inc} \\ x'_{b0} &= x'_b - z_b \tan \theta_{inc} \end{aligned}$$

次に荷物の優先度を p とし, これらの交点を矩形から円の方向に $p : 1 - p$ で内分した点から伸ばした, 傾きが $1/\tan \theta_{inc}$ の直線が最適位置・角度直線 $f_{opt}(x')$ となり, 式は以下の通りになる.

$$f_{opt}(x') = \frac{x' - (x'_{u0} - p(x'_{u0} - x'_{b0}))}{\tan \theta_{inc}}$$

4.5.3 最適位置・角度をサーボの角度に変換

求めた最適位置・角度直線 $f_{opt}(x')$ を実現する各サーボの角度を求める. 第1サーボの角度は雨の方角と同じであるため, $\theta_{dir}[\text{rad}]$ となる. 第2サーボの角度は xyz 空間における第2サーボの位置 $(0, -0.14, h - 0.08)$ を用いて計算する. これらは実際にユーザと第2サーボの位置関係を測定した結果に基づいた値である. 第2・3サーボ間の長さは 0.27m であるため, 第2サーボの稼働範囲は xyz 平面において中心座標が $(0, -0.14, h - 0.08)$, 半径が $0.27[\text{m}]$ の上半球として表せる. これを $x'z$ 平面に射影すると, 半径は変わらず中心座標が $(-0.14 \cos \theta_{dir}, h - 0.08)$ の上半円になる. この上半円と $f_{opt}(x')$ の交点を求め, 交点と上半円の中心を結んだ直線の z 軸に対する角度を $\pi/2[\text{rad}]$ から引いたものが第2サーボの角度となる. 第3サーボの角度は降雨方向と同じ角度であるため, $(\pi - \theta_{inc})[\text{rad}]$ から第2サーボの角度を引いた値となる.

なお, 第2サーボは角度によっては駆動部がデバイスやユーザの頭部に干渉する場合がある. そのため実際に角度による干渉具合を確認した結果より, 第2サーボの角度を 40° 以上に制限した. この角度以上であれば, 駆動部は姿勢を正した状態のユーザの頭部やデバイスに対して過度に干渉しないと判断した.

4.6 駆動

計算した角度を基に実際に各サーボを動作させる. 第2, 第3サーボは, Raspberry Pi における高度な GPIO 制御を可能とするライブラリである pigpio を用いて, ハードウェア PWM 制御により角度を変更する. この際, pigpio の仕様によりパルス周期に対するパルス幅の比を 100 万倍にした値を設定する必要がある. 目的の角度が $\theta_{180}[^\circ]$ の時のパルス幅 $w_{2,3}[\mu\text{s}]$ は, 第2, 第3サーボが $500\mu\text{s} \sim 2500\mu\text{s}$ の範囲で角度を指定する可動域 180° のサーボであるため, 以下の式で求められる.

$$w_{2,3} = 500 + 2000 \times \frac{\theta_{180}}{180}$$

この値を 100 万倍にしたものを pigpio を用いて設定することで, 目的の角度に第2, 第3サーボを設定することが出来る.

一方で第1サーボは角度を指定して動作させることは出来ず, PWM 制御により回転速度のみ調整することが出来る. その為, Reliabrella では第1サーボの角度の読み取りをロータリーエンコーダとフォトセンサを用いて外部から行い, 目標角度になるタイミングで回転を停止させて第1サーボの角度を調整している. ロータリーエンコーダには第1サーボの回転を 2 倍にして伝えることで, 第1サーボが 1 回転するごとに 100 パルスが $1/4$ 周期ずれて 2 つ出力される. この 2 つのパルスの立ち上がりとしち下がりそれぞれ見ることで, 1 回転を 400 分割する精度で角度変化を得ることが出来る. ただし, これでは絶対角度が得られないため, 第1サーボが 0° の時にフォトセンサが反応するようにすることで, フォトセンサが反応する度に角度を 0° に設定し絶対角度を得ることが出来る. 第1サーボの PWM 制御においては PCA9685 と呼ばれるサーボドライバを用いる. このサーボドライバはパルス周期を 4096 分割した時のパルス幅を指定するため, パルス周期が $20000\mu\text{s}$ の第1サーボに $w_1[\mu\text{s}]$ のパルスを送りたい場合に設定する値 w_{sig} は以下の式で求められる.

$$w_{sig} = 4096 \times \frac{w_1}{20000}$$

なお, 第1サーボの停止パルス幅は $1500\mu\text{s}$ であるので, 停止の際にはサーボドライバに 307.2 の値を設定する.

これらのサーボを瞬時に目標角度に設定すれば環境の変化に対する応答性は向上するが, 高速で動作させると強い慣性によりユーザへの負担に繋がる. そのため Reliabrella では, 各サーボの回転速度を毎秒 90° に制限した.

4.7 BLE 通信

Raspberry Pi では, BLE 通信を行うために BLEPeripheral を実装する. 実装は, Node.js のライブラリ bleno を用いて行う. Node.js の version は 7.4.0, npm の version は 6.14.4 を使用した. これは, 後述の Node.js のライブラリ

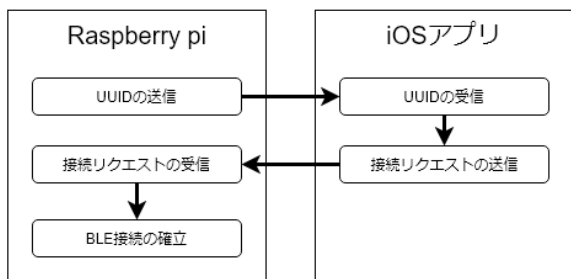


図 11: 通信の流れ

python-shell に対応しているためである。今回使用したものの中から主要な Node.js のライブラリ名及びライブラリの version を表 1 に示す。

表 1: 使用した Node.js のライブラリ

ライブラリ名	ライブラリの version
bleno	v0.5.0
bluetooth-hci-socket	v0.5.1
python-shell	v1.0.8
util	v0.12.3

実際の通信の流れを図 11 に示す。Reliabrella の起動時に Raspberry Pi から UUID の送信が行われる。送信された UUID を操作用アプリで受信し、操作用アプリから Raspberry Pi へ接続リクエストの送信を行い、Raspberry Pi と操作用アプリの接続を確立する。

4.8 傘の制御

傘の制御は Node.js のライブラリ python-shell を用いて実装する。これは、操作用アプリからユーザの身長、荷物の位置・優先度、降雨強度を用いて Raspberry Pi に送信した値をもとに Python プログラムを実行するためである。Python プログラムでは、操作用アプリから受信した値に加え、風速計、風向計から得た雨の角度をもとに傘の最適位置・角度を計算し、傘の制御を行う。

4.9 操作用アプリ

操作用アプリは iOS アプリケーションとして Xcode にて開発を行った。Raspberry Pi との通信には BLE を使用し、操作用アプリをセントラル、Raspberry Pi をペリフェラルとして接続する。図 12 はアプリ起動時の画面（左）であり、右にスワイプすると身体の向きを変更できる。アプリ画面に表示されている円はユーザが所持している荷物であり、ドラッグ操作でユーザが装備している部分に移動する。画面下のユーザの身長、優先度、降雨強度の項目をそれぞれ入力し Raspberry Pi に値を送信する。送信はアプリ画面右下のボタンをタップすることで行う。送信された

値と最適位置・角度計算を参照し、プログラムが実行される。ペリフェラルの識別には UUID を使用する。荷物の位置は原点を足元、右横方向を x 軸、進行方向を y 軸、上方向を z 軸として座標を小数で送信する。プログラム実行中にアプリから入力した値を破棄、又は変更したい場合は画面右下のボタンをタップすることでプログラムが停止状態に移行する。値を変更したい場合は再度入力を行い、画面右下のボタンをタップすることで値が更新された状態でプログラムが実行される。図 14 は実際にアプリを実行した時に前方から風が吹いているとした際の駆動であり、図 13 の値を入力としている。

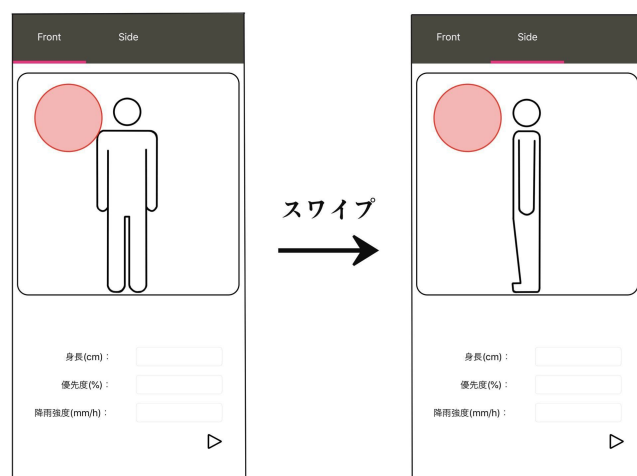


図 12: アプリ画面の遷移

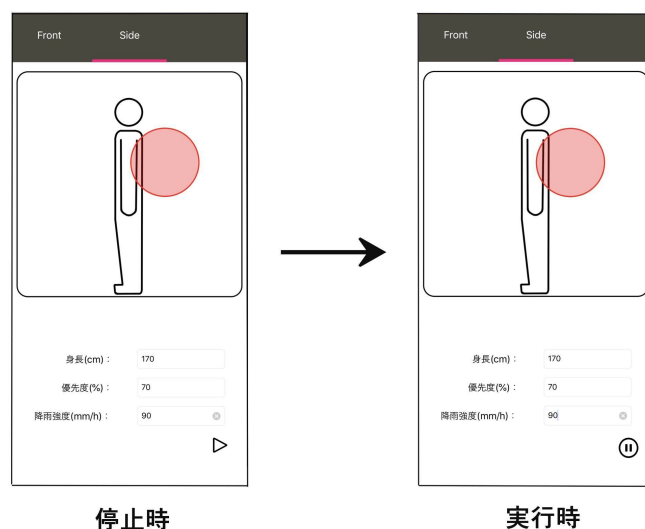


図 13: アプリの停止時と実行時

表 2: 評価結果

	1 回目 (pixel)	2 回目 (pixel)	3 回目 (pixel)	4 回目 (pixel)	5 回目 (pixel)	平均 (pixel)	割合 (%)
停止時	11115	2507	6184	7138	5829	6554.6	100
動作時	4203	1816	2610	2053	1211	2378.6	36.3
手動時	760	684	354	547	2343	937.6	14.3

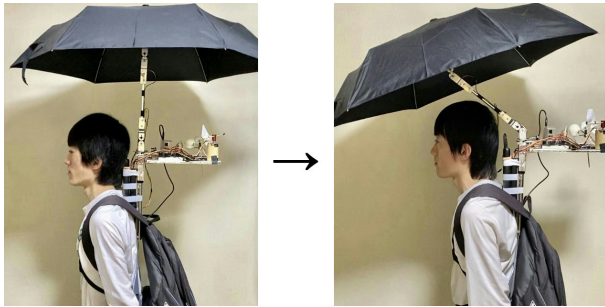


図 14: 実際にアプリを実行した時の駆動

5. 評価・考察

本章では、Reliabrella を実際に使用した際の防雨性能を評価した結果とその考察について述べる。評価は 5 月 10 日の 7 時から 9 時の 2 時間に静岡県湖西市で行い、降雨強度は 0.5mm であった。この時、降雨強度は気象庁が発表する 1 時間降水量の予想を参考にした。なお評価実験を行った当時は、流行していた COVID-19 の影響により各都道府県に緊急事態宣言が発令されていた。それにより評価において多人数が関与したり、人通りのある街中などで評価を行うことが困難であったため、評価は風により最適位置・角度に傘を動作させる効果に焦点を絞った簡易的なものとした。

5.1 評価方法

評価は、「Reliabrella 停止時 (以下停止時)」「Reliabrella 動作時 (以下動作時)」「Reliabrella に使用したものと同サイズの傘をユーザ自身が動かした場合 (以下手動時)」の 3 パターンを比較して行う。なお、動作時は優先度を 0 とし、ユーザの身体のみ守る設定にして行い、停止時は Reliabrella の第 1 サーボを 0° 、第 2 サーボと第 3 サーボを 90° の状態で固定して行った。評価においては、無風状態で 10m の距離を分速 80m の速さで歩いて移動した際、胸部に張り付けた 10cm 四方のシートにどれだけ雨滴が付着するかの検証を各パターン 5 回ずつ、計 15 回行う。これにより、理論上ではユーザに対して前方から約 1.33m/s の風が吹き付ける状態となる。歩行時の速度である分速 80m という値は、「不動産の表示に関する公正競争規約施行規則」[21] において定められた徒歩所要時間に基づいたものである。シートを張り付ける箇所を胸部に選んだ理由としては、胸部が足元から傘の頂点の中間付近に位置している

ことや、評価環境では基本的に雨が前方から吹き付けるために、背面と比べ前面により多くの雨滴の付着が想定されることなどがある。

5.2 評価結果

評価実験においては各評価毎にシートの写真を撮影した。その写真を全てサイズが 300×300 px となるように調整し、雨滴付着部が黒となる 2 値画像に変換した上で、黒色のピクセルの数をカウントしたものを結果とする。実際の結果を表 2 に示す。なお、参考として各パターンからそのパターンの平均値に結果が近い 2 値画像を図 15, 16, 17 に示す。

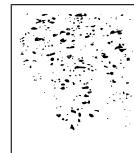


図 15: 停止時

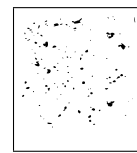


図 16: 動作時

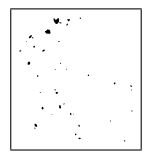


図 17: 手動時

5.3 考察

実験結果から、Reliabrella の優位性は停止時より高く、手動時より低いことが分かる。これより、Reliabrella における最適位置・角度は多少なりとも効果があると考えられる。しかし、Reliabrella では風速・風向を検知してから傘の角度の変更を行うため、検知してから傘が雨に対する最適な角度に変更するまでの間に時間を要する。そのため、手動時と比較すると風に対する応答性に欠け、歩き始めや風の吹き始め、またその終わり際に雨に濡れやすくなると考えられる。

6. 展望

今回の評価においては、Reliabrella は手動時より優位性が低い結果となってしまった。Reliabrella ではユーザの負担軽減のためにサーボモータの動作速度を制限している他、風向・風速計自体の応答性の悪さも相まって、手動時と比較すると環境の変化に対して素早く反応することが出来ない。他にも、ユーザは気象予報などを参照して降雨強度を入力する必要があるため、ユーザの手間の多さや降雨強度の変化に対するリアルタイム性などに関して課題が残っていると見える。更には、最適位置・角度自体の計算にも未

だ大きな問題点が残っている。駆動部は3つのサーボモータで構成することで角度だけでなくある程度の位置も調整することが出来るが、位置の調整は第1サーボが向いている方向に限られる。例えば正面から雨が吹き付けている場合に、その方向へ傘を傾けたまま横方向へ移動させることが出来ない。今回の評価実験では荷物が無い状態において前方向から吹き付ける雨のみ想定していたが、ユーザの身体から大きく飛び出るような荷物を持った場合などは、雨の方向によっては防雨性能が著しく低下すると思われる。この問題は第2・第3サーボ間に360°サーボモータを1つ追加すれば解決するが、今回は加工技術や耐久性の観点から断念せざるを得なかった。また、Reliabrellaは1人で装着することがやや難しく、その点が使用に際するハードルとなり得る。その様な防雨性能以外の点も今後改善していきたい。

参考文献

- [1] Ojoie: 知ってほしい傘の歴史!! 洋傘の歴史とは!!、Ojoie(オンライン), 入手先 <http://www.ojoie.com/blog/知ってほしい傘の歴史-洋傘の歴史とは/> (参照 2020-05-18).
- [2] JUPA: 洋傘の歴史, JUPA(オンライン), 入手先 <http://www.jupa.gr.jp/pages/history> (参照 2020-05-18).
- [3] JUPA: 傘の種類, JUPA(オンライン), 入手先 https://jupa.gr.jp/pages/umbrella_shubetsu (参照 2020-05-18).
- [4] 株式会社プラネット: 意識調査 From プラネット, 株式会社プラネット(オンライン), 入手先 https://www.planet-van.co.jp/shiru/from_planet/vol86.html (参照 2020-05-18).
- [5] アサヒパワーサービス: free Parasol, アサヒパワーサービス(オンライン), 入手先 <https://www.it-service.co.jp/freeparasol/index.html> (参照 2020-05-18).
- [6] A'DESIGN AWARD & COMPETITION: Umbrella Coat Raincoat, A'DESIGN AWARD & COMPETITION(オンライン), 入手先 <https://competition.adesignaward.com/design-image.php?y=2011&design=23642> (参照 2020-05-18).
- [7] Lieben: ハット! アンブレラ, 日傘と傘の専門店リーベン(オンライン), 入手先 <https://www.kasa-higasa.com/SHOP/3810-1.html> (参照 2020-05-18).
- [8] YANKO DESIGN: Phone-brella, YANKO DESIGN(オンライン), 入手先 <https://www.yankodesign.com/2015/10/15/all-weather-texting/> (参照 2020-05-18).
- [9] Wantis: Wantis Creative UFO Waterproof Hands Free Umbrella Rain Hat Headwear Cap School Pranks Performance Costum, Amazon(オンライン), 入手先 <https://www.amazon.com/dp/B074MDSK3D/?tag=incred08-21> (参照 2020-05-18).
- [10] Nubrella: nubrella WEATHER PROTECTION.REINVENTED, Nubrella(オンライン), 入手先 <https://www.nubrella.com/> (参照 2020-05-18).
- [11] THANKO: アタマンブレラ, THANKO(オンライン), 入手先 <https://www.thanko.jp/shopdetail/000000003084/> (参照 2020-05-18).
- [12] EuroSchirm: Swing hadsfree, EuroSchirm(オンライン), 入手先 https://euroschirm.com/schirm/Swing_hadsfree/index.cgi?session=uqTYmPaWl6ZKM&sprache_land=deutsch (参照 2020-05-18).
- [13] BASHERRY: 背負う傘 手で持たず 両手が完全にフリーになる 折りたたみ式, Amazon(オンライン), 入手先 <https://www.amazon.co.jp/dp/B07TDPKZLL> (参照 2020-05-18).
- [14] Lazo-Antunez, R. and Andrade-Madriñan, B. R.: Automated Hands-Free Umbrella, US 9,603,422 B1, 2017-03-28.
- [15] Shinji, K. and Masaaki, F. and Tsutomu, H.: Eyeglass-based hands-free videophone, Proc. ISWC'13, pp.117-124, ACM (2013).
- [16] Takashi, M. and Sho, H.: Pileus Internet Umbrella:Tangible Mobile Interface of A Lovely Umbrella, Proc. TEI'09, pp.41-42, ACM (2009).
- [17] 岡島厚, 内藤玄一, 藤谷徳之助ほか: 風を測る-計測器・計測システム-, 日本風工学会誌, Vol.1999, No.42, pp.58-108 (1999).
- [18] 翁長謙良, 吉永安俊, 趙 廷寧: 身長と歩幅の関連に関する一考察-学生の歩測の事例から, 琉球大学農学部学術報告, No.45, pp.149-155 (1998-12-01).
- [19] Chand, I. and Bhargava, P. K.: Estimation of Angle of Deflection of Rain at Driving Rain Prone Stations in India, JAABE, Vol.4, No.1, pp.223-229 (2005).
- [20] 河内まき子・持丸正明, 2005 AIST 人体寸法データベース, 産業技術総合研究所 H16PRO 287.
- [21] 不動産公正取引協議会連合会: 不動産の表示に関する公正競争規約施行規則, 入手先 https://www.sfkoutori.or.jp/webkanri/kanri/wp-content/uploads/2019/01/h_sekouk-isoku.pdf (参照 2020-05-18).