

令和2年度 修士論文

RGBカメラとDepthセンサを用いた生体情報センシングと
その応用

指導教員 弓仲 康史 准教授

群馬大学大学院理工学府 理工学専攻
電子情報・数理教育プログラム

情報通信システム分野第3研究室 弓仲研究室

T191D044 ZHOU JUNKAI

目次

目次.....	2
第1章 緒言	3
第2章 Depth センサを用いた頭部姿勢センシング	7
2.1 頭部の姿勢の座標系	7
2.2 点群レジストレーションと ICP.....	8
2.3 頭部姿勢センシング	9
2.4 データ表示と疲労提示.....	12
第3章 RGB による目の状態センシング.....	13
3.1 視線変化センシング	13
3.2 PERCLOS	17
第4章 生体情報センシングアプリケーション.....	18
4.1 開発環境とアプリケーションの概要	18
4.2 機能検証	22
第5章 結言	25
参考文献	26
謝辞.....	29
研究業績	30

第1章 緒言

ドライバーの警戒心の低下は、交通事故の大きな原因となっており、世界中で深刻な問題となっている。2019年に中国統計局が公表したデータによると、2019年には、原付以上の交通事故が159,335件あり、43,413人が死亡している[1]。特に疲労運転が交通事故の20%以上を占めると言われており、実際には、この比率がさらに大きくなっている。また、日本の警察庁により公表されたデータによると[3]、2019年に、原付以上の死亡事件数が図(1-1)で示している通り、3,099件あり、その内、酒酔い運転や疲労運転または漫然運転などの、ドライバーの警戒心の低下により発生した事件は1,887件となっている。

◎ 原付以上運転者（第1当事者）の法令違反別死亡事故件数の推移

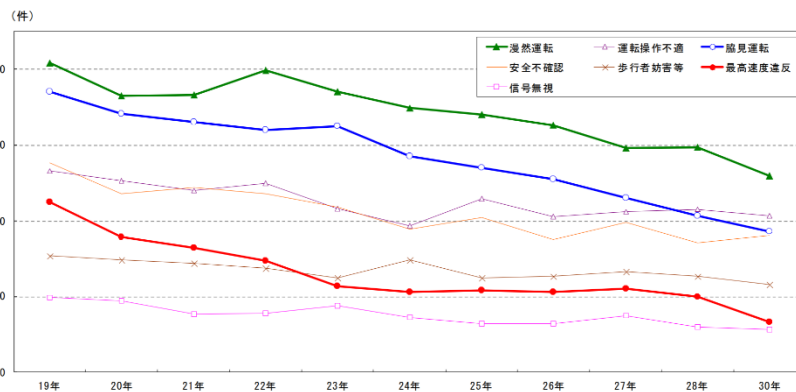


図 1-1 法令違反別死亡事件の推移

そのため、疲労運転または漫然運転など、ドライバーの警戒心の低下を検出することを目的としたドライバーモニタリング技術が注目されている。ドライバーモニタリング技術は、主に三種類の情報を用いており、具体的には、ドライバーの生理パラメータ、ドライバー上半身の特徴、および運転時の車両のパラメータである。ただし、運転時の

車両のパラメータの信頼性が不足していることから、通常は、補足的な判断根拠として使われることが多い。また、ドライバーの生理パラメータによる判断は、精度は高いが、センサを直接ドライバーの皮膚に接触させて測定する必要がある。したがって、カメラや非接触センサを用いたドライバーの上半身の特徴に注目した生体情報センシングが望まれている [2]。

ドライバーが運転する時、周辺の状態を確認するために、頭部が上下左右に移動する。例えば、対向車や歩行者、信号や標識などを確認するために、運転中に頭部が移動する。これに対し、居眠り運転などでは頭部動作が少なくなったり、上下に動くことなどが想定される。本研究は、このような頭部の動きのセンシングを目的としている。これにより、ある指定した時間範囲で、頭部の動作がない場合、ドライバーに居眠り等の注意を提示することができる。

頭部の姿勢をセンシングするために、距離情報を取得可能な Depth センサを用いた ICP (Iterative Closest Point) という技術を用いる。文献 [4] の IVVI2.0 プロジェクトでは、ICP を用いた頭部姿勢センシングの応用が提案されている。各フレームの間で ICP を行い、計算した各フレーム間の姿勢を累積し、現時点の頭部の姿勢を計算する有効なセンシング方法である。しかし、ICP で計算した姿勢変換には、誤差が含まれているため、計算した姿勢に対して、測定時間により誤差が累積され、測定結果に影響を及ぼす。さらに、頭の姿勢が大きく変化すると、計算が不能になる。そこで、本研究の解決方法

として、第二章で、事前変換を導入する。事前変換を導入することで、各フレーム間の変換の累積が不要となるため、累積誤差が存在しない利点がある。

次に、ドライバーが周辺の状況を確認する時は、常に頭部の姿勢が大きく変化するわけではなく、代わりに視線が大きく変動するため、本研究では、RGB カメラで視線変化のセンシングを姿勢変化の補足としている。これらは、第三章で詳細に説明を行う。また、顔のセンシング画像から、PERCLOS と呼ばれている閉眼の時間比率の測定も実現した。

本研究は、Intel が 2020 年に発売した RealSense L515 をセンサとして用いた。この Depth カメラは LiDAR 技術を用いて、図(1-2)で示しているように物のエッジの部分と角度急が変化している部分などで、Azure Kinect の ToF 方式より高い測定精度が実現可能である [6][7]。この角度変化耐性により、本研究で提案する事前変換を導入し

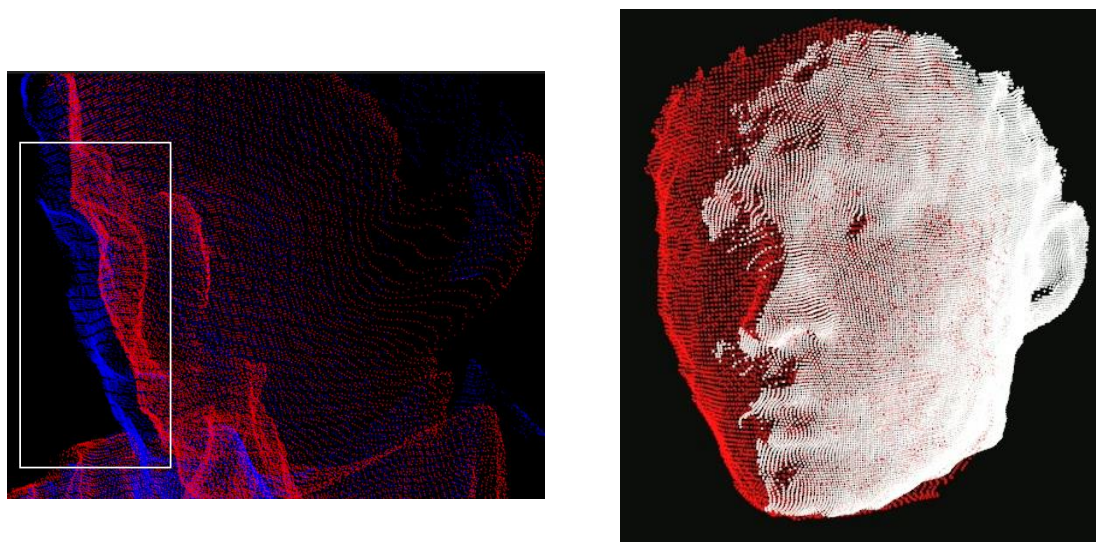


図 1-2 左：Kinect の ToF 式が角度により、精度が変わる。よって、頭の形が角度により、変わる。右：L515 の LiDAR 式は角度が変化しても、雑音が少ない。

た ICP の計算効率を高めることが可能となる。

本論文は以下のように構成される。

第一章は緒言であり、本研究の背景と目的、及び、概要を述べる。

第二章では、頭部姿勢のセンシングの原理と実現方法を説明する。

第三章では、目の開閉の特徴のセンシング方法の検討と実現方法を紹介する。

第四章では、開発したアプリケーションの検証結果について述べる。

第五章は結言であり、本論文をまとめる。

第2章 Depth センサを用いた頭部姿勢センシング

本章では、Depth センサを用いた頭部姿勢センシングの原理を説明する。まず、姿勢の座標系の定義と表示方法を説明し、本研究で使ったセンシング方法を述べる。次に、従来の問題点と解決策を説明し、データの表示方法を示す。

2.1 頭部の姿勢の座標系

本研究では、人間の頭部を剛体として扱う。したがって、測定開始前に、頭部の初期位置を指定する必要がある。頭部の姿勢は、現在位置と初期位置との回転角と平行移動で表現可能となる。

ここで、回転は、座標系の三つの軸を中心とした回転行列で記述できる。

$$R_z = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_y = \begin{bmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & \sin(\rho) \\ 0 & -\sin(\rho) & \cos(\rho) \end{bmatrix} \quad (\text{式 2-1})$$

R_z は z 軸を中心とした回転で、roll と呼ばれている角を表示している。そして、 R_y は yaw と呼ばれ、y 軸を中心とした回転角である。 R_x は x 軸を回った回転で、pitch と呼ばれている(図 2-1)。

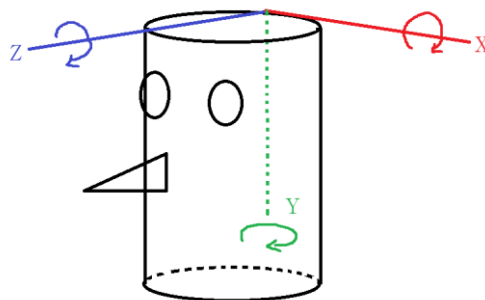


図 2-1 青矢印は Roll、赤いは Pitch、緑色のは Yaw である。

本研究では、この三つの回転行列を $R_z * R_y * R_x$ の順番で掛け算した結果 R で頭部の回転を記録する。

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = R_z * R_y * R_x \quad (\text{式 2-2})$$

そして、この回転行列 R から、以下の式で三つの回転角を計算することが可能である。

$$\text{Roll} = \arctan\left(\frac{R_{21}}{R_{11}}\right) \quad \text{Yaw} = \arctan\left(\frac{-R_{31}}{\sqrt{R_{32}^2 + R_{33}^2}}\right) \quad \text{Pitch} = \arctan\left(\frac{R_{32}}{R_{33}}\right) \quad (\text{式 } 2-3)$$

次に、平行移動は三次元ベクトルで表示でき、 $[t_x \ t_y \ t_z]^T$ で表せる。

簡略化するために、同次座標で前述の回転と平行移動を表示する。以下のようになる。

$$H = \begin{pmatrix} R & t \\ \mathbf{0}^3 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{式 } 2-4)$$

本研究では、頭部姿勢のセンシングは、式(2-4)で表した同次座標の行列を求めることに対応する。

2.2 点群レジストレーションと ICP

本節では式(2-4)の変換行列を算出する原理を説明する。

点群は三次元空間の点の集合であり、物体の三次元サンプリングともいえる。そして、点群のレジストレーションとは、一つの物体が式(2-4)によって位置と方向が変わった後、変化する前の点群と変化した後の点群の間の対応点ペアを用いて、式(2-4)の解を求めることである。

例えば、鼻の頂点 p_{bef} が頭の運動と共に回転や並進した後 p_{aft} になっていたとすると、

$$p_{bef} = H * p_{aft} \quad (\text{式 } 2-5)$$

対応する式(2-5)の H を計算することが点群レジストレーションである。

三次元の点の場合では、一つの対応点ペアで三つの方程式を作れる。式(2-4)の中で、12個の変数があるため、対応点ペアが四つあれば、変換行列を算出可能となる。四つ以上があれば、対応点ペアが含まれている雑音を改善することが可能である。

したがって、以下の式で最小値を取る時、最優変換行列を算出可能である。

$$\Sigma^2 = \sum_n \|p_{aft} - H * p_{bef}\|^2 \quad (\text{式 } 2-6)$$

この式を最小化するために、最急降下法と、文献[12]で提出された対応点の間の共分散行

列を SVD 分解する方法の二種類がある。SVD の方法のほうが、計算量が少ないため、よく使われている。

頭部の姿勢センシングの難点は、式(2-6)の計算ではなく、対応点のペアを確定することである。

例えば、建物など、著しい特徴点が多くある時、通常 Harris や SIFT などのアルゴリズムを用いて特徴点の対応点ペアを探すことが可能である。しかし、人間の頭部の特徴は顔に集中しており、光や角度の変化により、特徴点として検出できなくなってしまう。これに対し、近年の顔のランドマーク技術の発展により、顔のランドマーク検出が一つの有効な対応点ペアリングとされている。しかし、顔ランドマークが少なく、頭が横に向くと、図(2-2)のように、誤検出が増え、消えてしまう可能性が高い。



図 2-2 横に向くとランドマークが間違ふ。

ゆえに、ICP(Iterative closest point)を使う必要がある。

ICP アルゴリズムは、一気に対応点ペアを正しく検出して変換行列を計算するのではなく、ある時点の最近点をペアとして計算し、繰り返して最優変換に近づける方法である。具体的には、ICP が直接その場合で距離の近い点をペアとし、式(2-6)を計算した後、算出した変換を初期点群に適用し、再び最近点をペアとして計算する。初期点群を、現在の点群に近づける効果がある。図(2-3)はこのアルゴリズムの流れを示している。

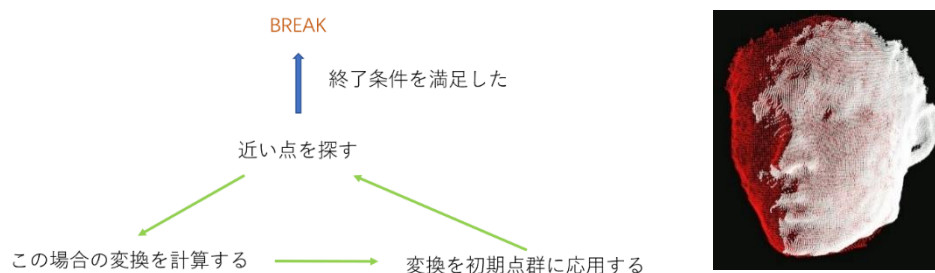


図 2-3 左:ICP の流れ。右:ICP を応用した結果。

2.3 頭部姿勢センシング

ICP を用いて、頭の初期位置から現在位置までの変換が計算可能であったが、ICP のペア

リング方式により、誤変換になる可能性が高くなった。

これは人間の頭の形が球体に近いためである。市販の depth センサの精度が有限であり、鼻や唇などの、頭を球体と区別できる特徴が、雑音により弱められることが原因と考えられる。

そこで、解決方法として、ICP の計算の前に事前変換を適用する方法を提案する。この事前変換により、初期位置の頭部の点群を一つ大体の方向に変換できる。この事前変換は、二つのベクトルで計算することが可能である。

人間の顔では、ほとんどの画像情報が目、鼻と口の領域にある。そして、市販の depth センサの精度により、目の部分の点群が雑音とは区別不能となっている。よって、この二つのベクトルが鼻と口の領域にある。

本研究で用いたのは、二つの口角から鼻の頂点までに引いた図(2-4)に示すベクトルである。

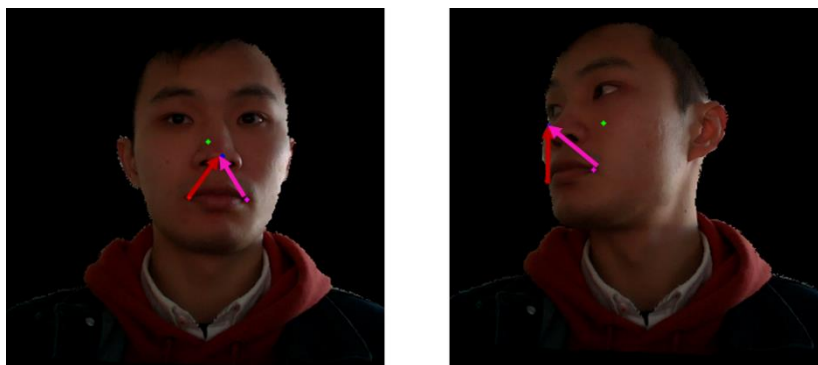


図 2-4 ベクトルの例。

二つの口角と鼻の位置は MTCNN という RGB 画像による顔認識のニューラルネットワークを使って検出することが可能である。ただし、このネットワークは RGB 画像で動作するものであり、図(2-5)で示しているように鼻の位置で微小な誤差を含んでいる。二次元の点では、この誤差が小さいが、三次元の点の場合、誤差が無視できなくなる。

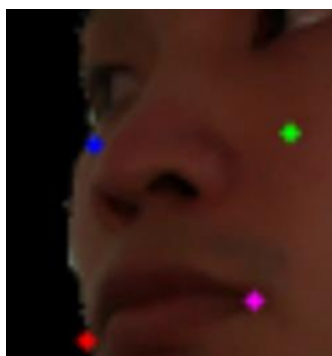


図 2-5 鼻の頂点認識が間違った例。

そのため、MTCNN で検出した後、再確認する必要がある。簡単な方法では、鼻の点の

周辺 3*3 の領域の中で最もカメラに近づく点を鼻の点とする。または、3*3 の領域共計 9 点をカメラとの距離で近から遠くの順番で並び、真ん中の三つの点の座標の平均値を修正した鼻の位置とする。この方法で、Depth センサの雑音を抑えることが可能である。

次に、二つのベクトルの回転は、図(2-6)が示しているように、その二つのベクトルの外積を軸とした回転に見直すことが可能である。

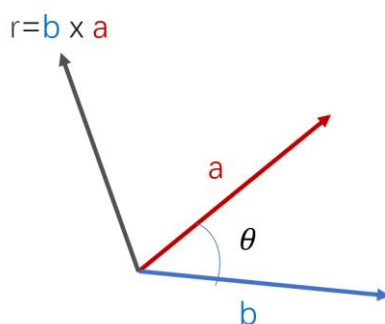


図 2-6 ベクトル回転の例。

この外積 r と角度 θ は eigen ライブラリの toRotationMatrix() 関数を使って回転行列に変換することが可能である。

また、平行移動は回転を応用した初期点群口角の座標と現在点群の口角の座標の差を取ることにより計算できる。式(2-3)に対応する計算を示す。

$$t = p_{now_mouth} - R * p_{init_mouth} \quad (式 2-7)$$

最後に、初期点群と現在点群が共に二つベクトルを持っているため、四つの変換を計算できる。この中で、最もよい変換を選ぶ必要がある。本研究で用いた方法は、四つの変換すべてを初期点群に適用して、その後、現在点群と最近点ペアを探す。最近点ペア同士の間距離の平均値を、その変換の評価基準とすると、この平均値が小さければ小さいほど、その変換が良いと考えられる。

このような方法で、図(2-7)中の青点群を、緑色の点群に変換することによって、ICP がローカルな最小値への落ちこみを防止可能となる。

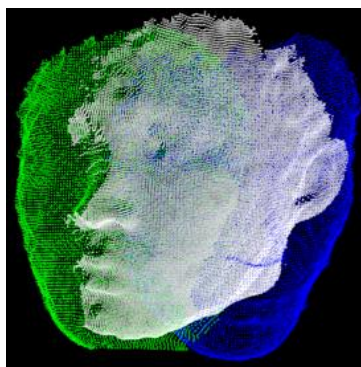


図 2-7 青:初期点群。白:現在点群。緑:事前変換を応用した初期点群。

2.4 データ表示と疲労提示

以上の ICP により、頭部の姿勢を計算できるが、平行移動の部分は本研究の対象ではないため、回転の部分のみを使う。式(2-3)により、回転行列が三つの角に変換できる。ドライバーが眠気をもよおすと pitch が低くなることが測定できる。

そして、ドライバーが疲労している時、注意意識が低下し、周辺状況の確認回数が減少する。この現象を角度で表示すると、角度の変化の期間が長くなることに対応する。

しかし、2.1 節で説明した三つの角度では、この現象を検出しにくいので、2.2 節のベクトルの回転角を使ってもよい。Eigen ライブラリの `fromRotationMatrix()` 関数により、ベクトルの回転角に変換可能である。

または、点群の雑音が少ない場合は

$$\theta = \arccos \left(\frac{v_{bef} \cdot v_{aft}}{\|v_{bef}\| \|v_{aft}\|} \right) \quad (式 2-8)$$

で計算してもよい。

角度変化がない時間区間の長さの制限は、一つのタイマーで実現できる。本研究で設定した長さは 15 秒である。タイマーは常に動いており、時間が 0 になったら赤いウインドウを表示し疲労の警告提示をする。ただし、角度の変化を検出できた場合、15 秒にリセットする。このような方式で、ドライバーが 15 秒間で周辺を確認したら、疲労検出の再開が実現できる。

しかし、周辺状況の確認は、常に頭部の動きだけではなく、目線の変化で実現する。このため、頭の姿勢だけでは、不十分である。第三章では、RGB カメラで目線の変化と目の閉じる比率の検知を検討する。

第3章 RGB による目の状態センシング

ドライバーの頭部の姿勢だけでは、疲労状態の判定が困難なため、本研究では目の状態を補足根拠として用いる。目の状態として、目線の変化と目が閉じる時間比率二つを検出する。測定は頭の姿勢がベクトルの回転角で 10° 未満の時だけで行う。まず、目線の変化センシング方法を説明し、次に、閉じる比率に注目した PERCLOS の算出方法を検討する。

3.1 目線変化センシング

目線の変化は本研究で補足根拠として使われ、ドライバーの三次元目線ではなく、目線の変化だけを分析できるものである。具体的には、ドライバーの目線が左から右に移動したなどが本研究の検出対象である。

目線の変化は、次の3ステップで検出できる。

- Step 1. 顔検出し、顔の画像から両目の画像を抜き出す。
- Step 2. 両目の画像で目の動点と定点の位置を検出する。
- Step 3. 二つの定点から動点までに引いたベクトルを作り、ベクトルを足し算する。



図 3-1 角度修正。

ステップ1では、第二章で述べた MTCNN を用いている。そして、目の位置を確定するのは dlib というライブラリである。直接顔の画像から各 ROI のエントロピーで目を探すの



図 3-2 抜き出した目の例。

も一つの有効の方法だが、運転席のドライバーの光線の変化が不安定で、エントロピーの方法の成功率が著しく減少する。これに対して、dlib が顔のランドマークを検出できる。しかし、図(2-2)で示した通り、顔が横に向くと検出不可能となる。本研究で頭の姿勢が 10° 未満の場合だけ利用するため、図(2-2)のようにならず。また、顔の画像を抜き出す前に、図(3-1)のように、両目のインナーコーナーで画像の角度を修正する必要がある。このように、目周辺の特徴点を使い、図(3-2)で示している通りに、目の画像を抜き出すことができる。

ステップ2では、定点は目のインナーコーナーを使う。アウターコーナーより、インナー

コーナーが安定しているためである。

動点は黒目の中心を使う。この中心は黒目の位置を表示し、真の中心ではない。この仮中心を用いて、黒目のエッジを検出し、楕円フィッティングすると、真の中心を確定できるが、計算量が著しく増大してしまう。本研究の対象は、目線の変化だけであるため、精度の高い中心の確定は行わない。

仮中心の確定には、図(3-3)で示している積分投影関数と分散投影関数を使う。

グレースケール画像の中、座標が (x,y) の点の数値が $I(x,y)$ とすると、積分投影関数は以下のようにとなっている。

$$IPF_v(x) = \frac{1}{y_2 - y_1} \sum_{y_1}^{y_2} I(x, y) \quad (式 3-1)$$

$$IPF_h(y) = \frac{1}{x_2 - x_1} \sum_{x_1}^{x_2} I(x, y) \quad (式 3-2)$$

ここで、 y_1, y_2, x_1, x_2 を指定する必要がある。 $IPF_v(x)$ が垂直積分投影関数で、 $IPF_h(y)$ が水平積分投影関数と呼ばれている。垂直積分投影関数の例としては、画像が皮膚→目白→目黒→目白→皮膚の順番で変化しているため、目黒の部分が最も低くとなっており、周辺の目白とでもピクセルの変化が激しいので、積分投影関数で簡単に検出できる。

分散投影関数は、画像の変化に注目した関数であり、以下のようにとなっている。

$$VPF_h(y) = \frac{1}{y_2 - y_1} \sum_{y_1}^{y_2} ||I(x, y) - IPF_v(x)|| \quad (式 3-3)$$

$$VPF_h(y) = \frac{1}{x_2 - x_1} \sum_{x_1}^{x_2} ||I(x, y) - IPF_h(y)|| \quad (式 3-4)$$

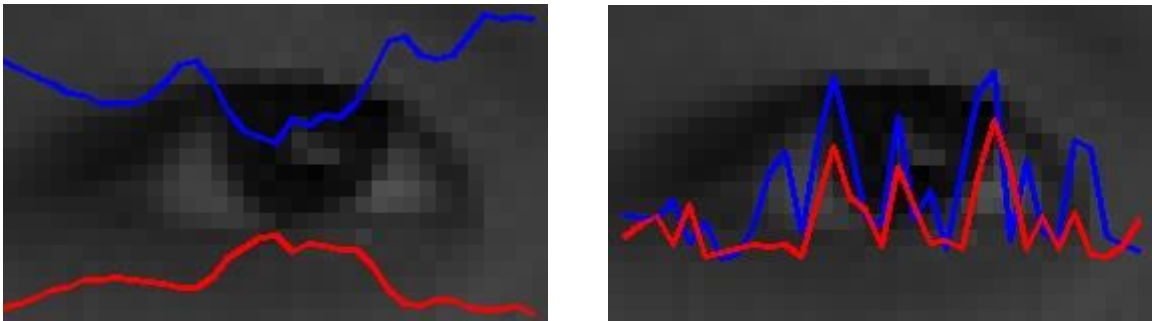


図 3-3 左:垂直投影関数の例。青は積分投影、赤は分散投影。右:その微分。

図(3-3)で示した通り、垂直投影関数の微分を分析することで、黒目の仮中心を確定することが可能である。具体的には、絶対値を取った微分関数の一番目と二番目のローカル最大

値の座標の平均値を仮中心の x 座標とする。同じ方法で、 y 座標も計算できる。

ただし、 x 座標を計算する時、光線により、垂直積分投影関数が図(3-4)のように、間違
う可能性が高くなる。

このため、本研究は x 座標を計算する時は垂直分散投影関数を使っている。式(3-1)と式
(3-3)中の y_2 と y_1 の区間が目画像の広さと同じである。

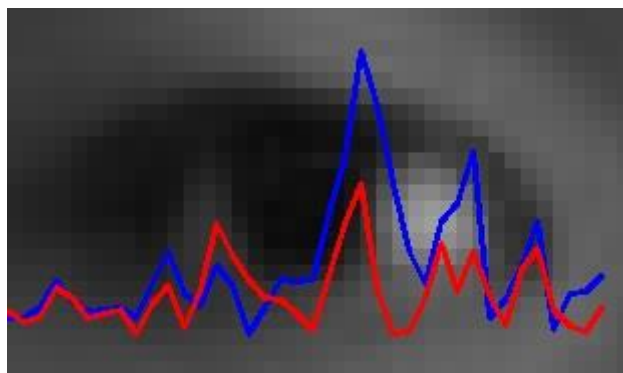


図 3-4 積分投影関数が失敗した例。

次に、 y 座標を確認する時は、水平積分投影関数を使っている。垂直より水平方向の変化
が少ないため、黒目が黒いという特徴の方が信頼性が高い。式(3-2)の水平積分投影関数中
の x_2 と x_1 が x 座標を計算する時、二つのローカル最大値の x 座標である。

また、図(3-5)左で示した通り、虹彩の反射が分散投影関数に強い影響を与えているため、
消去する必要がある。反射の光は周辺の黒目は違っているため、ガウシアンフィルタを使っ
て消去可能である。しかし、ガウシアンフィルタで反射を消去することにより、エッジの情
報が著しく弱くなり、不要な雑音を導入してしまう。

このため、ガウシアンフィルタをかけた画像から、再び処理する必要がある。本研究での
対策は、元画像とフィルタした画像の重み付け平均値を使うことである。 $I(x,y)$ が元画像の

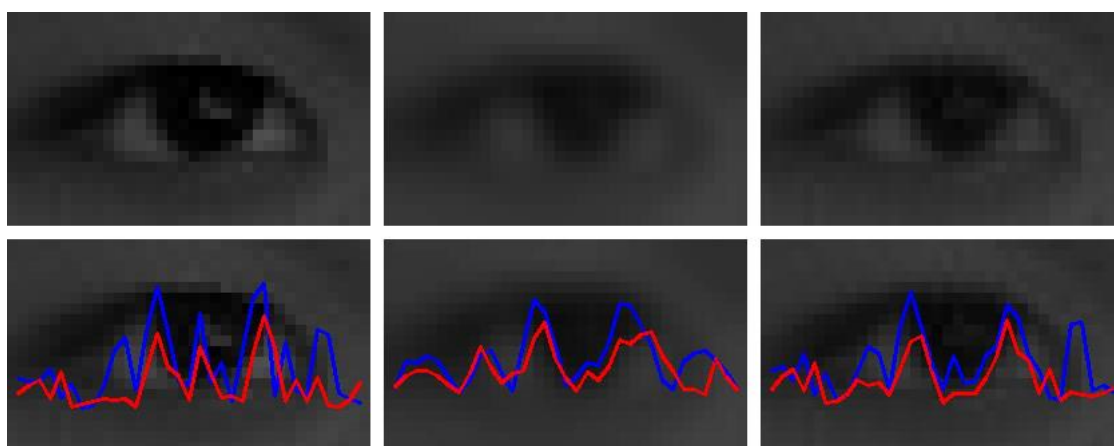


図 3-5 左:元画像。中:フィルタをかけた画像。右:修正した画像。

ピクセル値で、 $I'(x,y)$ がフィルタした画像のピクセル値とすると、修正した画像のピクセル
値 $I_{fix}(x,y)$ が以下の式で表示できる。

$$I_{fix}(x, y) = p * I(x, y) + (1 - p) * I'(x, y) \quad (式 3-5)$$

中で、 p は[0:1]範囲内の重みであり、以下の式で計算する。

$$p = e^{\left(\frac{-d^2}{\delta^2}\right)} \quad \text{where } d = I(x, y) - I'(x, y) \quad (式 3-6)$$

δ は分散であり、大きければ大きいほど、元画像と近づく、修正した画像の輪郭がより分かりやすくなる。本研究で試した結果、 $\delta = 15$ の時、一番効果が得られた。

以上の方法で目の動点が検出できる。

ステップ3では、ステップ2の時検出できた定点から動点まで引いたベクトルを作り、足し算すると、目線の方向がわかる。すなわち、人間が左に見ると、左眼の黒目が左に行き、左眼のインナーコーナーと離れ、左のベクトルが長くなる。右目の黒目も左に行くが、右目のインナーコーナーに近づき、右目のベクトルが短くなる。このような原理で、二つのベクトルを合成すると、目線の方向がわかる。

また、このベクトルはドライバーとカメラの距離変化で長さも変わるため、二つのインナーコーナーの距離で標準化する必要がある。

$$V_{normalized} = \left(\frac{x}{d}, \frac{y}{d}\right) \quad (式 3-5)$$

中で、 d はインナーコーナーの距離である。

3.2 PERCLOS

一定期間内の目が閉じる時間と全体時間の比率が一つ有効な疲労判断根拠と考えられている。この比率は PERCLOS と呼ばれており、以下の式で定義される。

$$PERCLOS = \frac{N_{eye\ closed}}{N_{all}} * 100\% \quad (式 3-5)$$

ここで、 N はフレームの数である。

本研究では、目が閉じている判定は、前節のステップ 1 で使った目の長方形の長さとおきの比率を使う。

$$CloseRatio = \frac{width}{height} \quad (式 3-6)$$

目を閉じると、この比率が著しく増大する。よって、あるしきい値を指定し、このしきい値を超えた場合、目が閉じているとする。しかし、閉じる時の比率が人により違い、各自設定する必要がある。本研究の場合はしきい値として3を用いた。

最後に、15 秒の間で、閉じる時間の比率を目の閉じる比率とする。本研究で試した結果、疲労しない状態では、10%程度にある。

第4章 生体情報センシングアプリケーション

本章は本研究で開発したドライバーの状態検知を目的とした生体情報センシングアプリケーションの概要とその検証の結果を示す。

4.1 開発環境とアプリケーションの概要

開発環境は以下のようになる。

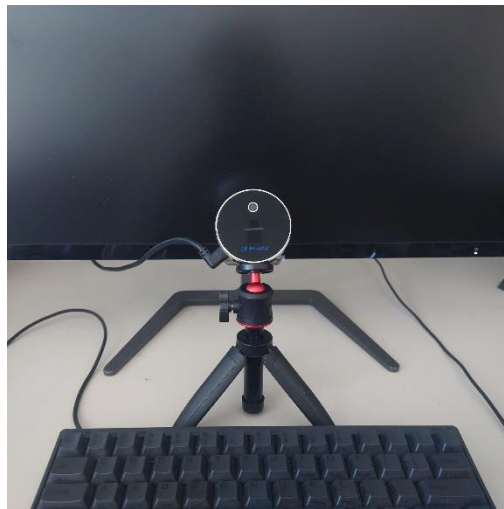


図 4-1 Intel RealSense L515

ハードウェア	
CPU	Intel Core i7-8700K
Graphics Board	GeForce GTX1050
Camera	Intel RealSense L515(図 4-1)
ソフトウェア	
言語	C++
ライブラリ	OpenCV 4.2 PCL 1.8.1 Eigen Dlib libtensorflow

本研究は 2.3 節の事前変換を実現する時、OpenMP を用いてマルチスレッドで実現しているため、4 コア以上の CPU が必要となる。

開発されたアプリケーションは図(4-2)のようになっている。



図 4-2 アプリケーションの様子

左の黒い部分はセンシングしている人の様子と鼻の頂点と口角の位置を表示している。

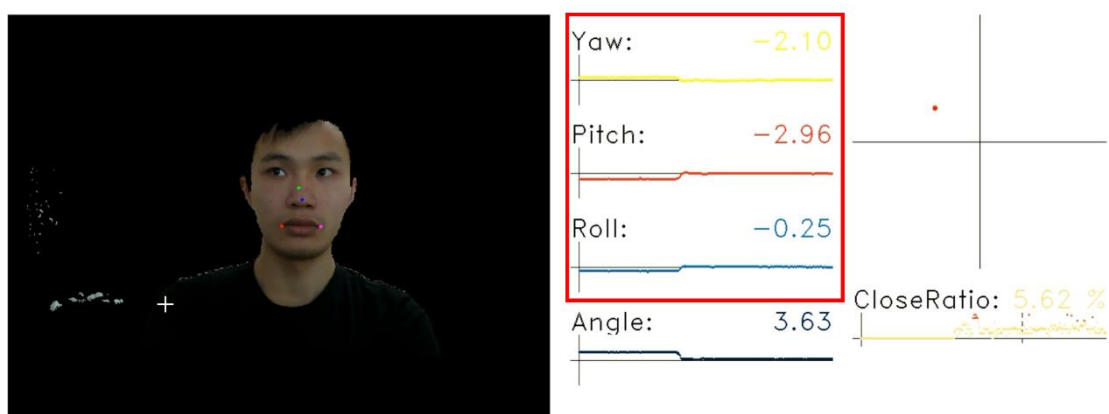


図 4-3 オイラー角のグラフ

図(4-3)の中、赤い長方形で囲まれた部分は 2.1 節で説明した、三つの軸を中心とした回転角、オイラー角と呼ばれている角度である。単位は度(°)である。デフォルトでは、上下限は 90° と設定している。



図 4-4 ベクトル回転角

図(4-4)の中、標記されたのは 2.3 節で紹介したベクトルの回転角である。回転の方向を考えず、純粋な角度であり、単位も度(°)である。

最後に、図(4-5)の右にあるのは 3.1 節の目線変化の部分と 3.2 節の閉眼時間比率を示すグラフである。右上は、3.1 節で紹介した両目のベクトルを合成した目線のベクトルである。単位は 1 両目インナーコーナー距離である。右下は、3.2 節で説明した閉じる時間比率である。数字は閉じる時間対全体時間の比率である(式 3-5)。グラフは式(3-6)の目の長方形の比率を点の形式で表示している。横軸は時間で、縦軸は比率である。ただし、目が完全に閉じる時、比率が大きくなり、上限を超える可能性が高いため、グラフで表示不可能になってしまう。そこで、グラフで表示するための座標は以下の式で 1 以下に抑えている。

$$y_{in\ graph} = 1 - e^{-(CloseRatio-t)} \quad (式\ 4-1)$$



図 4-5 目線変化と閉じる時間比率

ここで、CloseRatioは式(3-6)で計算する。tは点の集中程度を制御している。大きければ大きくほど、点が分散する。tが大き過ぎで、 $y_{in\ graph}$ がマイナスになることもある。繰り返し試した結果、t = 2の時の表示効果が一番良い。

また、各パラメータの調整のため、図(4-6)のようなコントロールパネルを設定した。

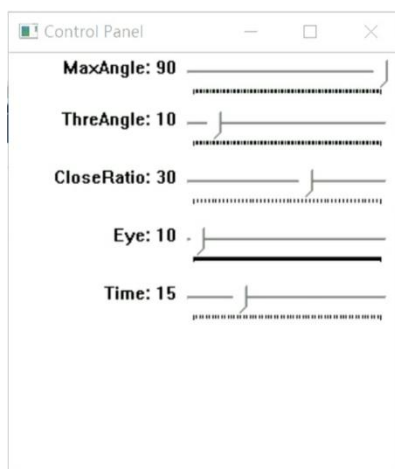


図 4-6 コントロールパネルの様子

MaxAngle はオイラー角のグラフの上下限を設定できる。ThreAngle は第三章の角度のしきい値を設定できる。頭部の姿勢の角度が、このしきい値未満の場合のみ、目の測定を実行する。CloseRatio は 3.2 節の閉眼判定のしきい値を設定できる。このしきい値を超えた時、目が閉じていると判定し、グラフで赤い点で表示する。本研究の検証では 30 を使っている。

Eye は目線のベクトルの変化のしきい値である。目線のベクトルが移動した距離が、このしきい値を超えた時、2.4 節のタイマーをリセットする。本研究の検証では 17 である。Time はタイマーの初期値を設定できる。

4.2 機能検証

本研究で開発したアプリケーションは測定開始前に、顔の初期位置を指定する必要がある。

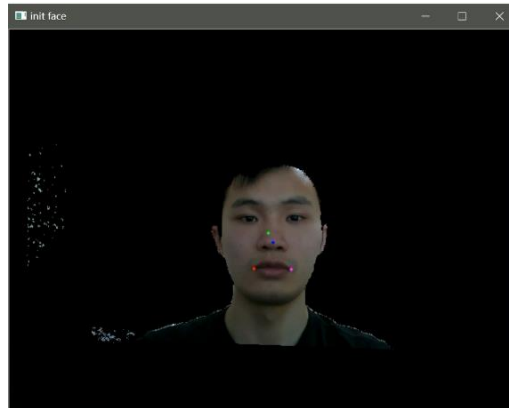


図 4-7 初期位置指定

図(4-7)の画面で ESC を押すと、測定に入る。

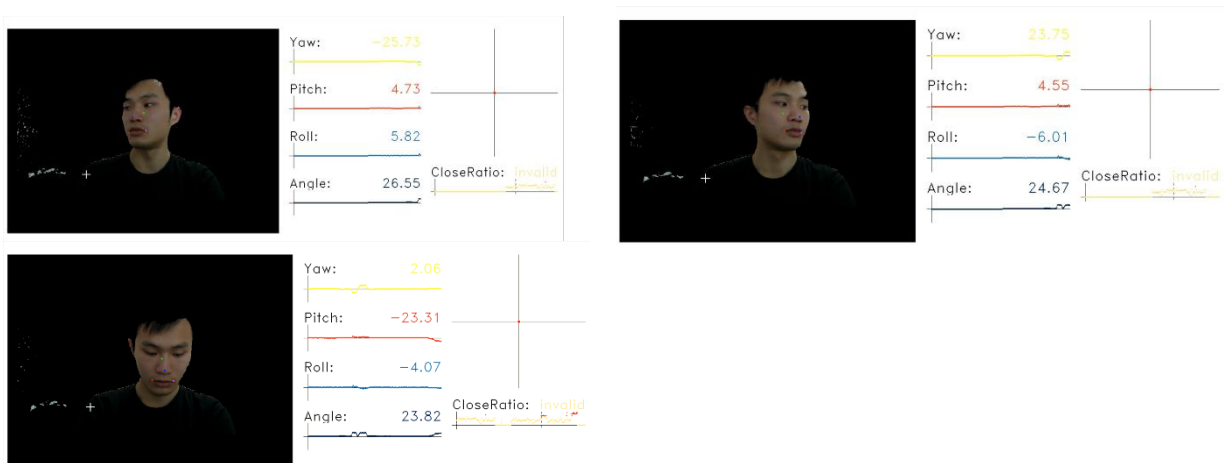


図 4-8 姿勢センシングの例

図(4-8)で示している通り、頭が左右に動くと Yaw がそれに応じて変化することがわかる。人が完璧に左右に動くわけではないため、Pitch と Roll も微小な変化をした。また、Angle の数値が Yaw に近づいているが、常にプラスとなる。これは、この角度が回転の方向を考えていないためである。方向が図(2-6)の外積 r にある。

```

time raimin: 14.812
time raimin: 14.601
time raimin: 14.357
time raimin: 14.059
time raimin: 13.98
time raimin: 13.642
time raimin: 13.338
time raimin: 13.036
time raimin: 12.712
time raimin: 12.432

```

図 4-9 タイマーの例

また、CMD ウィンドウズでは、図(4-9)のように、タイマーの残り時間を表示している。

```

time raimin: 15
eye:0.0953766

time raimin: 14.918
eye:0.0197287

time raimin: 14.844
eye:0.0420779

time raimin: 14.632
eye:0.0867952

time raimin: 14.527
eye:0.232619

time raimin: 15
eye:0.362883

time raimin: 15
eye:0.492302

time raimin: 15

```

図 4-10 タイマーがリセットされた例

図(4-10)第五行目のように、目線または頭のベクトル回転角が大きく変化したのを検知した場合、タイマーをリセットする。従って、ドライバーが定期的に周辺の状態を確認できれば、タイマーがゼロにならない。

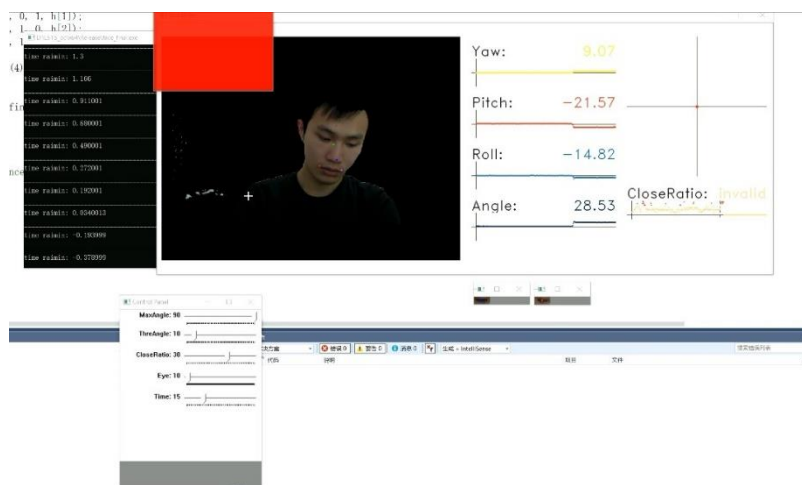


図 4-11 警告提示する例

一定期間内、周辺の状態を確認していないと、タイマーがゼロまたはマイナスとなり、本研究の場合は、図(4-11)のように、赤いウインドウを表示し、警告提示する。



図 4-12 閉眼時間比率

最後に、図(4-12)が示している閉眼時間比率の部分では、前 15 秒のデータを使って計算している。つまり、図(4-13)で示した点線の右の部分である。この比率が正常の時は 10% ぐらいにある。この数字に大きく超えた場合は要注意である。



図 4-13 長時間目が閉じている例

第5章 結言

本研究では、ドライバーが運転時、周辺の状態を確認する頻度を着目点とし、ドライバーの頭部の姿勢と目の特徴をセンシングし、姿勢と視線の変化の頻度によってドライバーの状態を検知し、異常時には警告を提示するシステムを提案した。また、ICPが頭部姿勢センシングの時、ローカル最小値に落ちやすい問題点の改善方法を新たに開発した。ICPの方法により、頭部姿勢センシングを実現し、投影関数を用い、視線変化も実現した。ただし、このアプリケーションはまだプロトタイプであり、実際の運転環境で実験し、タイマーや閉眼時間比率の期間などの具体的な数字を調査する必要がある。また、実際に疲労状態での、運転データの取得は危険であるため、VRヘッドセットを利用した仮想空間での実験が望ましいと考えられる。

参考文献

[1]. National Bureau of Statistics of China

<http://www.stats.gov.cn/english/>

[2]. 田村僚士、Web カメラと低廉なデバイスを用いたドライバーの運転状況モニタリング

システム、群馬大学工学部電子情報理工学科、情報システム分野第三研究室弓仲

研究室、卒業論文、2017

[3]. 警察庁

<http://www.npa.go.jp/publications/statistics/koutsuu/toukeihyo.html>

[4]. Gustavo A. Peláez C., F. García, A. Escalera, J. Armingol, Driver Monitoring Based on

Low-Cost 3-D Sensors, IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 4, August 2014

[5]. R. Grace, V.E. Byrne, D.M. Bierman, J.-M. Legrand, D. Gricourt, B.K. Davis, J.J. Staszewski, A drowsy driver detection system for heavy vehicles, 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference, 1998.

[6]. Intel® RealSense™ LiDAR Camera L515 User Guide

https://support.intelrealsense.com/hc/article_attachments/360081846174/L515_

[User_Guide_v1.0.pdf](#)

[7]. H. Sarbolandi, D. Lefloch, A. Kolb, Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect, Journal of Computer Vision and Image Understanding May 21, 2015

[8]. OpenCV

<https://opencv.org/>

- [9]. Point Cloud Library,PCL

<https://pointclouds.org/>
- [10]. 高橋一輝、深度センサを用いたプロジェクションマッピングとその医療福祉応用、群馬大学大学院理工学府理工学専攻電子情報・数理教育プログラム、情報システム分野第三研究室弓仲研究室、修士論文、2015
- [11]. X. Gao, T. Zhang, Y. Liu,Q. Yan, 14 Lectures on Visual SLAM: From Theory to Practice, Publishing House of Electronics Industry, 2017.
- [12]. K. S. ARUN, T. S. HUANG, S. D. BLOSTEIN, Least-Squares Fitting of Two 3-D Point Sets, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. Pami-9, No. 5, September 1987.
- [13]. N. Zhang, H. Zhang, Driver Fatigue State Detection Based on Facial Key Points, ICSAI 2019.
- [14]. Z. Zhou, X. Gengm, Projection Functions for Eye Detection, Pattern Recognition Society Volume 37, Issue 5, P1049-1056, 2004.
- [15]. Q. Wang, J. Yang, M. Ren, Y. Zheng, Driver Fatigue Detection: A Survey, Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006
- [16]. S. Shi1, W. Tang1, Y. Wang, A Review on Fatigue Driving Detection, ITA 2017.
- [17]. 福田祥吾、HMD の内部センサを活用した関節可動域および頭部動揺測定システム、群馬大学理工学部電子情報理工学科、情報システム分野第三研究室弓仲研究室、卒業論文、2019

- [18]. Oculus, Corneal Sphere Tracking for Generating An Eye Model, U.S.Patent
10,109,067. 2017-03-10

謝辞

本論文は筆者が群馬大学大学院 理工学府 理工学専攻 電子情報・数理教育プログラム 情報通信システム分野 第三研究室 弓仲研究室で行った研究についてまとめたものである。

本研究を進めるにあたり、日頃よりご指導いただきました弓仲康史准教授に深くお礼を申し上げます。また、主査の三輪空司准教授、副査の本島邦行教授、また、副指導員の高橋俊樹准教授に適切なご指導を賜りましたことを感謝いたします。

最後に、日頃よりご助言・ご意見を頂いた弓仲研究室の皆様には感謝の意を表し、謝辞とさせていただきます。

令和3年3月

研究業績

ZHOU JUNKAI、弓仲 康史、“RGB カメラと Depth センサを用いた生体情報センシングとその応用,” 第 11 回電気学会東京支部栃木・群馬支所合同研究発表会, 2021 年 3 月 (オンライン)