

ディスプレイを用いて光電脈波センサに 任意の脈波を計測させる手法の提案

藤井 敦寛^{1,a)} 村尾 和哉^{1,2,b)}

概要: 脈波センサは緑色の LED を皮膚に照射して、血管を通して反射した光の変化から脈波を計測する光電式容積脈波記録法 (PPG) と呼ばれる方式のものが一般的であり、スマートウォッチをはじめとする市販のウェアラブルデバイスに導入されている。脈波センサは機構の特性上、データの取得に血流を必要とするが、義手やウェアラブルロボットアームなど人工的な身体にスマートウォッチを装着する場合、血流が存在しないため正しいデータが取得できない。そこで、ディスプレイを用いて光電脈波センサに任意の脈波データを計測させる手法を検討する。本手法が実現すれば、身体と義手の接合部などで計測された脈波を入力することで、その値を義手に装着したスマートウォッチに読み取らせることが可能となる。本稿では心拍数に注目し、目標とする任意の心拍数を入力することでディスプレイを制御し、ディスプレイ上に装着したスマートウォッチで目標とする心拍数が取得できるか調査した結果について述べる。ディスプレイ描画プログラムとスマートウォッチアプリケーションを実装し、スマートウォッチと 2 台のディスプレイを使用して評価実験を行った。その結果、目標心拍数とスマートウォッチで計測された心拍数の誤差が Display A で平均 1.8 回、Display B で平均 1.6 回であり、全体で 3 回以内と高い精度で心拍数を再現できた。

1. はじめに

健康管理への意識の高まりから、自身の生体情報を記録するウェアラブルデバイスが広く普及している。記録する生体情報は活動量や呼吸数、体温、心電位、血圧、視線などさまざまな情報があり、脈波や心拍数もそのひとつである。脈波や心拍数を取得するために用いられる脈波センサは、赤外線や赤色、550nm 付近の緑色波長の光を LED で皮膚に照射する。動脈の血中に存在する酸化ヘモグロビンがこれらの光を吸収する特性があり、心拍のタイミングで血流量が増加することで反射光の量が減ることを利用して、フォトランジスタを用いて光量の変化を取得し、脈波を計測する。脈波データはこの反射光の変化の数値データであり、心拍数は脈波データに出現するピークを検出することで計測される。この脈波計測方式を光電式容積脈波記録法 (PPG: Photoplethysmogram) と呼ぶ。スマートウォッチをはじめとする市販のウェアラブルデバイスの多くは、この方式を利用した脈波センサを搭載している。脈波は生体情報の中でも重要なデータのひとつであり、脈波データを

用いて呼吸数を推定する手法を Havriushenko ら [1] が提案しているなど、脈波データを使用した研究は盛んである。

一方で当然ながら、義手やウェアラブルロボットアームなど人工的な身体には血流が存在しないため、スマートウォッチを生身の身体と同様に手首に装着しても生体情報を計測することはできない。通話やメッセージ、時計、決済などのスマートウォッチの機能や加速度センサや GPS などのセンサは人工的な身体であっても生身と同様にスマートウォッチを装着すれば利用できるが、脈波の計測はできない。脈波を計測するために足首に無理矢理スマートウォッチを巻くと、時計などの他の機能のユーザビリティが低下する。現状で脈波を計測したいのであれば、脈波センサを追加で計測可能な身体部位に装着し、スマートフォンなどと通信をしてデータを収集する必要がある。手首で体温計測を行う場面を考えても、店員と客がアクリル板で仕切られてアクリル板に空けられた穴越しにやりとりする場面では、人工的な身体を穴から差し出して計測してもらうことはできない。生身の身体向けに提供されている便利で使いやすいインターフェース、見た目に美しいデザイン、社会に浸透しているプロトコルなどを人工的な身体でもそのまま利用できるならば、そのような社会が望ましいと考える。

本研究では、ディスプレイを用いて光電脈波センサに脈

¹ 立命館大学大学院情報理工学研究科

² 科学技術振興機構さきがけ

a) atsuhiro.fujii@iis.ise.ritsumeit.ac.jp

b) murao@cs.ritsumeit.ac.jp

波データを計測させる手法を提案する。提案手法によって、義手やウェアラブルロボットアームなどの人工的な身体にスマートウォッチを装着する場合でも、身体と義手の接合部などで計測された脈波を入力し、手首に埋め込まれたディスプレイを制御することで、本人の脈波データをスマートウォッチに読み取らせることができる。さらに、スマートウォッチが提供する時計などの機能を生身の身体と同様に利用できる。また、人工的な身体にディスプレイを搭載するのみでスマートウォッチには手を加えないため、市販のスマートウォッチのデザイン、機能、重量などのさまざまな項目を比較して好きな機種を利用できる。このほか、遠隔地のロボットアバタに適用すれば、操作者の生体情報をアバタの身体でも計測できるようになる。本稿では、スマートウォッチに任意の心拍数を計測させることを目指す。

他方、提案手法の実現によって、光電脈波センサに任意の心拍数を計測させるのが可能であることが明らかになるため、悪意ある利用者が心拍数を偽って運動したふりをしたり、安静を続けているふりをすることもできると考えられ、光電脈波センサの脆弱性を明らかにする側面もある。提案手法を実現するデバイスが広く一般に実現可能となり、社会的影響が大きくなった場合は、現状の光電脈波センサの利用について議論する必要があると考えるが、本稿ではこの点については深く議論しない。

以降、2節で関連研究を紹介し、3節で提案手法の詳細を説明する。4節で提案手法の評価を行い、5節で今後の計画を述べ、最後に6節で本研究をまとめる。

2. 関連研究

本節ではウェアラブルデバイスのセンシング、スマートウォッチの利用、脈波データの利用に関する研究を紹介する。

2.1 ウェアラブルデバイスのセンシング

Ham ら [2] はスマートグラス用の入力デバイスとして、リストバンド型のデバイスを提案している。このデバイスはタッチパネルと慣性計測ユニットを搭載しており、タッチや手首をひねるなどのモーションで操作ができる。手首にデバイスを装着することで使用できるため、ユーザは動きを制限されず、自由度が高い。また、ポインティングにはタッチパネルを使用することで、入力の安定性を向上させた。Hernandez ら [3] は頭部装着型のウェアラブルデバイスである、Google Glass に内蔵された加速度センサ、ジャイロセンサ、カメラから脈拍数と呼吸数を認識する手法を提案している。Nishajith ら [4] は、視覚障害者の状況認識を支援するウェアラブルデバイスとして、スマートキャップの設計と実装を行った。デバイスは Raspberry Pi 3, Raspberry Pi NoIR Camera V2, イヤホン、電源から

構成される。Raspberry Pi NoIR (No Infrared) Camera V2 とは Raspberry Pi の赤外線カメラモジュールである。この赤外線カメラで得られる画像から検出された対象物について、イヤホンを通して音声で説明する。これらはいずれも身体部位に装着するウェアラブルデバイスに関する研究であり、様々な形状のデバイスを用いた研究が行われている。

さらに、デバイスの装着部位も多岐にわたる。Vahdatpour ら [5] は 25 人の被験者に頭部、胸部、両上腕、両前腕、腰部、両大腿部、両脛部の計 10 箇所に加速度センサを装着してもらい、日常行動下の加速度データを収集した。収集したデータから SVM (Support Vector Machine) を用いて、平均 89% の精度で装着部位を推定した。Szttyler ら [6] は 15 人の被験者の頭部、胸部、左上腕、左手首、腰部、ズボンの左ポケット、左足首の計 7 箇所に加速度センサを装着し、様々な身体活動における加速度データを収集した。収集したデータから Random Forest を用いて装着部位を推定し、平均 89% の精度を達成した。Kunze ら [7] は 6 人の被験者の右手首、右目付近の側頭部、ズボンの左ポケット、左胸のポケットの計 4 箇所に加速度センサを装着し、歩行動作におけるデータを収集した。収集したデータから C4.5 分類木を用いて装着部位を推定した。また、筆者ら [8] はウェアラブルデバイスで取得可能な生体情報である心電と脈波を利用し、特定の行動を装着者に行わせることなくウェアラブルデバイスの装着部位を推定する手法を提案している。

このように、ウェアラブルデバイスは様々な形状のものが提案されており、装着部位も広範囲であることから、活発な研究が行われている。

2.2 スマートウォッチの利用

ウェアラブルデバイスの中でもスマートウォッチは早くから市販化されていることもあり、多くの研究が行われている。Spinsante ら [9] は低強度の身体活動時にスマートウォッチから取得される心拍数に注目し、その精度を計測している。Sen ら [10] はスマートウォッチの加速度センサおよび、ジャイロセンサから得られるデータを使用して、手、箸、スプーンのいずれを使用して食べたかなどの食事行動を記録する手法を提案している。スマートウォッチ内蔵のカメラで食品画像を撮影し、画像識別を行うことにより食事内容も記録する。Johnston ら [11] はスマートウォッチは常に同じ場所、同じ方向で身に着けるものであることに着目し、スマートウォッチの加速度センサやジャイロセンサから得られるデータを使用して歩行に基づく生体認証を行う手法を提案した。一般的にスマートフォンはズボンのポケットやハンドバッグに格納して所持することが多い。これらの場所に比べ、スマートウォッチを装着する手首では活動の情報が大きく現れる傾向にある。Weiss

ら [12] は食事行動などの手に基づく身体行動において、スマートウォッチがスマートフォンよりも効果的に行動を識別できることを示した。“飲む”という行動において、スマートウォッチでは 93.3% の精度で識別できるのに対して、スマートフォンでは 77.3% の精度しか得られなかった。Iakovakis ら [13] はスマートウォッチを用いて、姿勢変化による血圧 (BP) の低下を予測することを目的とする研究を行っている。起立性低血圧 (OH) はめまいや失神を引き起こす可能性があるとしており、高齢者だけでなく若年層でも転倒のリスクがある。そこで、心拍変動データを取得することで起立性低血圧による転倒リスクを低減する数学的予測モデルを提案している。Mauldin ら [14] は市販のスマートウォッチから得られる加速度データを用いて転倒を検知する Android アプリケーション “SmartFall” を提案している。スマートウォッチは SmartFall を実行するスマートフォンとペアリングされており、SmartFall はデータのプライバシーを守りつつクラウドサーバと通信し、転倒の予測に必要な計算をリアルタイムに実行する。Ciabattoni ら [15] は様々な認知タスク中の精神的ストレスをリアルタイムに検出する手法を提案している。市販のスマートウォッチで取得したガルヴァニック皮膚反応 (GSR), RR 間隔, 体温 (BT) を用いてストレスを分類する。

人工的な身体ではウェアラブルデバイスを装着しても生体情報を収集できないため、これらの応用を利用できない可能性がある。加速度センサや GPS などのセンサを用いた手法は適用できるが、生体情報を利用する手法は適用できない。これに対して筆者らは人工的な身体であったとしてもウェアラブルデバイスの生体情報センサにデータを計測させることで、生身の身体と同様にこれらの応用できるようにすることを試みる。

2.3 脈波データの利用

Havriushenko ら [1] はニューラルネットワークを用いて脈波データから呼吸数の推定を行う手法を提案している。呼吸数を測定する場合には鼻腔内に設置した熱センサや伸縮性のある胸部ベルトなどを用いることが多い。しかしながら、これらの方法は睡眠の妨げとなる可能性がある。脈波データを使用する手法であれば、ウェアラブルデバイスへの実装が可能である。検証の結果、2.2 呼吸/分以下の平均呼吸数推定誤差を達成した。Han ら [16] はスマートウォッチから取得された脈波データから心房期外収縮 (PAC) および、心室性期外収縮 (PVC) を検出する手法を提案している。Goshvarpour ら [17] は単純な動的信号処理技術と統合手法から感情の変化を分類する手法を提案している。35 人の被験者から安静時および、特定の感情を刺激するような音楽を聴いている状態で心電図と指の脈拍を収集した。ポアンカレプロットを使用した後に SVM を用いて幸福、悲しみ、安らぎ、恐怖の 4 つの感情に分類した。Kajiwara

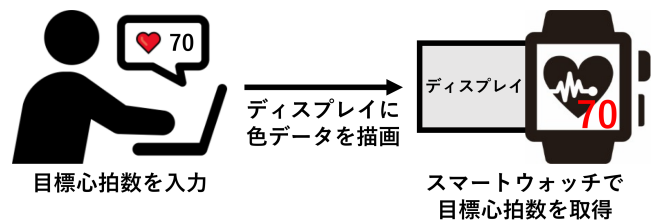


図 1 提案手法の概要

ら [18] は多くの物流企業が手動でのオーダーピッキングシステムを採用しており、感情やエンゲージメントが作業効率やヒューマンエラーに影響を与えることに注目し、ウェアラブルデバイスで取得した行動や脈波から、運動強度の高い作業中における感情やエンゲージメントを予測する手法を提案している。脈波、眼球運動、動作を深層ニューラルネットワークに入力することで、感情やエンゲージメントの推定を行う。検証実験の結果、誤り率 0.12 以下の精度で予測できることを明らかにした。Lee ら [19] は脈波データによる感情認識の速度向上に関する研究を行っている。行動価 (valence) と覚醒度 (arousal) に基づいた 2 次元の感情モデルを採用し、1D CNN (1 次元畳み込みニューラルネットワーク) を用いることで、1.1 秒の脈波データから感情認識を行う。1D CNN を二値分類 (行動価, 覚醒度が高いか低いか) として DEAP データセットを用いて実験したところ、行動価では 75.3%, 覚醒度では 76.2% の認識精度を達成した。

脈波データは身体の異常検知や感情推定ができるなど、重要な生体情報の一つである。市販のウェアラブルデバイスの脈波センサの多くは光電式容積脈波記録法を使用している。そのため、血流が存在しない人工的な身体にウェアラブルデバイスを装着する場合、脈波データは取得できない。筆者らは生体情報の中でも脈波データに注目し、人工的な身体でも生身の身体と同様の脈波データをウェアラブルデバイスに計測させる手法を提案する。

3. 提案手法

本節では提案手法の詳細を述べる。

3.1 概要

提案手法は任意の心拍数を設定すると、ディスプレイの表示が変化し、ディスプレイ上に装着したスマートウォッチに指定された心拍数を計測させる。提案手法の処理の流れを図 1 に示す。はじめに、ユーザはコンピュータに表示されているコンソールから取得したい心拍数を入力する。提案手法は入力された心拍数に応じてコンピュータに接続されたディスプレイの明暗を変化させる。そして、ディスプレイ上に装着されたスマートウォッチは入力された心拍数と同じ数値の心拍数を計測する。

3.2 心拍数の設定

本稿では提案手法によってスマートウォッチに計測させたい心拍数はディスプレイに接続されたコンピュータに明示的に与える設計としている。ただし、装着者の心拍数を別の心拍センサを利用して取得し、その数値を計測させたい心拍数としてリアルタイムに指定することもできる。設定された心拍数を H_{target} とする。

3.3 ディスプレイの制御

スマートウォッチが計測する心拍数が H_{target} となるように、ディスプレイの明暗を制御する。事前に、1回の脈拍をスマートウォッチに検出させることができるディスプレイの明暗の変化のデータ列 *Colors* を用意する。ヒューリスティックに探索した事前調査から *Colors* は以下の数値列とした。

$Colors = [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 250, 240, 232, 227, 225, 225, 229, 236, 245, 255]$

Colors はグレースケールのデータである。グレースケールとはコンピュータでの色の表現方法の一種であり、黒から白までの色の濃淡を 0 ~ 255 の 256 段階で表現する。具体的に *Colors* は以下の流れで生成している。

1. $Colors[i] = \sin\left(\frac{2\pi i}{19}\right)$ ($i = 0, \dots, 19$)
2. $Colors[i] = Colors[i] + 1$
3. $Colors[i] = 1$ (if $Colors[i] > 1$)
4. $Colors[i] = Colors[i] * 30 + 225$

Colors をプロットしたものを図 2 に示す。

光電式容積脈波記録法を用いた脈波センサは、赤外線や赤色、緑色の LED を皮膚に照射して、血管を通して反射した光の変化から脈波を計測する。脈拍が発生するタイミングで血流量が増加するため、血管でより多くの光が吸収され、反射する光は暗くなる。図 2 の値が低下している部分が反射光が暗くなる部分を表現している。グレースケールは値が小さくなるほど黒に近づく。黒は白に比べて光を吸収するため、ディスプレイが黒く描画されるほど、ディスプレイ上に装着したスマートウォッチから発せられてディスプレイを通して反射する光は暗くなる。

そして、1分間で *Colors* が H_{target} 回だけ再生されるように、*Colors* の各値の描画間隔 $T[s]$ を以下のように設定する。

$$T = 60 / (L_{Colors} * H_{target}) \quad (1)$$

ただし、 L_{Colors} は *Colors* のデータ長 (本稿では 20) である。得られた $T[s]$ ごとに *Colors* の値を 1 つずつ描画する。

3.4 ディスプレイ描画プログラムの実装

ディスプレイの明暗を変化させるプログラムを Python と Processing を用いて実装した。Processing^{*1}とは視覚的

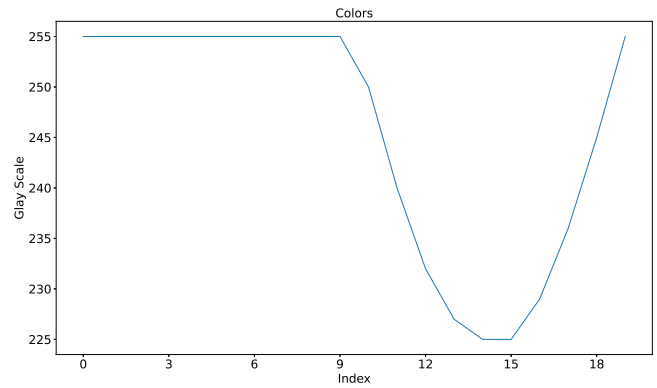


図 2 スマートウォッチに 1 回の脈拍を検出させることができるディスプレイの明暗の変化のデータ

な表現を得意とする、Java をベースに開発されたプログラミング言語であり、電子アートやビジュアルデザインの作成などに使用される。実装したプログラムの処理の流れを図 3 に示す。まず、Python で標準入力から目標とする心拍数 H_{target} を受け取る。色データの作成部分では Numpy^{*2}で *Colors* を作成する。そして、式 (1) を用いて $T[s]$ を計算する。作成した *Colors* を 1 つずつ Python の socket ライブラリを使用して Processing に送信し、描画の完了を待機する。Processing 側ではデータを受信すると、そのグレースケールを *background* メソッドを使用して、ディスプレイに表示されているウィンドウの背景色として描画する。描画が完了したら Python 側へ通知する。Python 側で通知を受信すると現在時刻を取得し、あらかじめデータを送信する直前に取得しておいた T_{now} と比較を行い、 $T[s]$ が経過すると次の色データを送信する。この流れを繰り返し、*Colors* のデータを全て送信したら、再び *Colors* の 1 番目から同様にデータを取り出して処理を行う。

4. 評価

本節では、提案手法の有効性を評価するために行った実験について説明する。任意の目標となる心拍数を与えた際にスマートウォッチで取得される心拍数を計測した。

4.1 スマートウォッチアプリケーション

評価実験ではスマートウォッチを利用して心拍数を計測するが、初期からインストールされているアプリケーションでは一定時間経過するとセンシングを停止して画面が OFF になったり、心拍数を数値として保存できないため、心拍数をロギングするスマートウォッチアプリケーションを実装した。

今回は評価実験に Google の Android をベースにスマートウォッチ向けに設計された OS である Wear OS by

^{*1} <https://processing.org>

^{*2} <https://numpy.org>

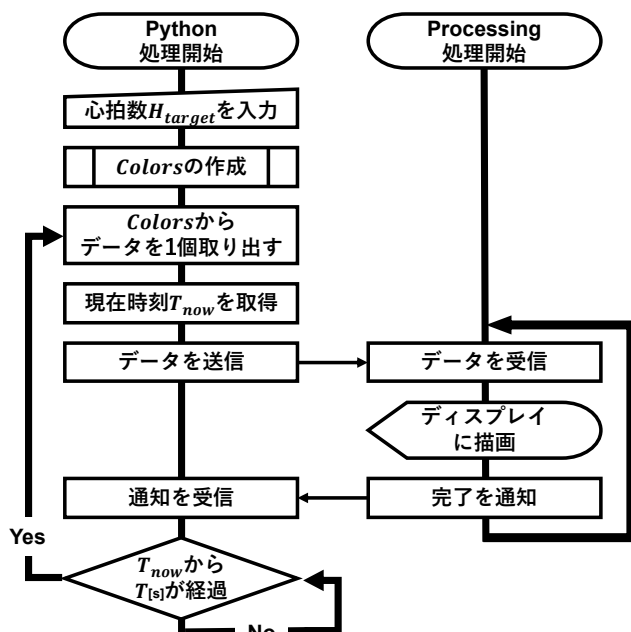


図 3 実装したプログラムの処理の流れ

表 1 実装に使用したセンサの詳細

取得対象	センサ番号	サンプリングパラメータ
心拍数	21	SENSOR_DELAY_UI
脈波データ	65572	SENSOR_DELAY_UI

Google^{*3}を搭載したスマートウォッチを使用するため、Android Studio で実装した。実装したアプリケーションを図 4 に示す。アプリケーションを起動すると、図中 (1) の画面が表示される。センサ値の取得が自動的に開始され、値に変化があった場合、(2) のようにセンサ値が表示される。“Heart” は心拍数センサ，“Pulse” は脈波センサの値を示す。データを記録する場合は“RECORD” ボタンを押下すると、(3) のように 60 秒間のキャリブレーションが開始される。このキャリブレーションは値の変動を安定させるために待機するものである。この間にスマートウォッチを固定しておく。60 秒間のキャリブレーションが終了すると、(4) のように 60 秒間センサデータを取得し、変数に保存する。センサデータの取得時間が終了すると、変数に保存しておいたデータを csv 形式でスマートウォッチのストレージに保存し、(5) のように取得完了を示すメッセージが表示される。実装に使用したセンサの詳細について表 1 に示す。脈波データを取得するためのセンサ番号はデバイスにより異なる。ここでは評価実験に使用した TicWatch Pro WF12106 (Mobvoi 社製) でのセンサ番号を示す。サンプリングパラメータの“SENSOR_DELAY_UI”とは、ユーザインタフェースの実装に適したサンプリングレートである。^{*4}

^{*3} <https://wearos.google.com>

^{*4} <https://developer.android.com/reference/android/hardware/SensorManager>

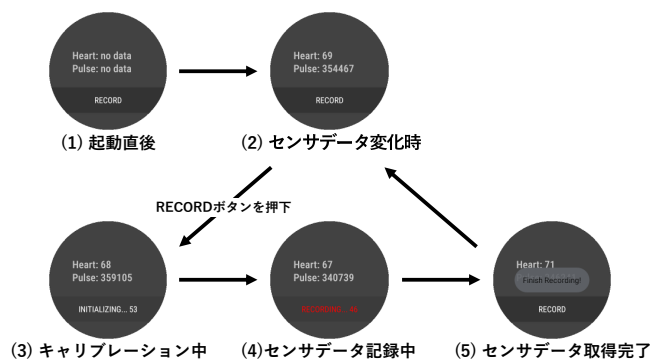


図 4 実装したアプリケーション

4.2 実験環境

3.4 節で実装したディスプレイ描画プログラムと 4.1 節で実装したスマートウォッチアプリケーションを使用してデータの収集を行う。スマートウォッチには TicWatch Pro WF12106 (Mobvoi 社製) を使用し、ディスプレイはプログラムの実行に使用するノート PC である Legion 7 15IMH05 (Lenovo 社製) の内蔵ディスプレイ (以下 Display A) とマイクロコンピュータである Raspberry Pi 向けに設計された小型の 3.5 インチディスプレイ (OSOYOO 社製) (以下 Display B) を使用した。プログラムを実行するノート PC と Display B の接続には HDMI を使用した。また、Display B には購入時に画面を保護するためのフィルムが貼り付けてあったが、剥がした場合は正しくセンサ値が取得できなかったことから貼り付けたままの状態で行った。心拍数データ取得のサンプリングレートは約 1Hz、脈波データ取得のサンプリングレートは約 21Hz である。

データの取得は次の流れで行った。まず、実装したスマートウォッチアプリケーションを起動する。次にディスプレイ描画プログラムを実行し、適当な心拍数でディスプレイの描画を開始する。その状態でスマートウォッチをディスプレイに設置し、センサデータが更新されていることを確認する。このときの状態を図 5 に示す。設置が完了したら、ディスプレイ描画プログラムの標準入力から目標心拍数を入力し、スマートウォッチアプリケーションの“RECORD” ボタンを押下する。60 秒間キャリブレーションした後に、60 秒間データ取得が行われる。目標心拍数を成人の平均心拍数である 60 回から 100 回 [20] まで 5 刻みで増加させながらデータを取得していき、100 回に達すると再度 60 回から取得を繰り返す流れを 3 セット行う。以上の流れでディスプレイ 2 台でデータの取得を行った。

4.3 結果と考察

設定した目標心拍数とスマートウォッチで取得された心拍数の誤差を表 2 に示す。この結果は 3 セットの平均値である。0 は目標心拍数と一致していることを示し、マイナスは回数が不足していることを示す。また、“Average” は



図 5 評価実験での心拍数の取得方法

表 2 設定した目標心拍数とスマートウォッチで取得された心拍数の誤差

目標心拍数	Display A	Display B
60	-1.0	-1.0
65	-1.3	-1.3
70	-1.0	0.0
75	-2.0	-1.7
80	-2.0	-2.0
85	-2.0	-2.3
90	-2.0	-2.0
95	-2.0	-1.7
100	-2.7	-2.3
Average	-1.8	-1.6

ディスプレイごとの誤差の平均値である。結果より、成人の平均心拍数である 60 回～100 回において、目標心拍数より 0～3 回以内の誤差でスマートウォッチに心拍数を入力できていることが確認できた。平均値ではあるが、目標心拍数を 70 回としたとき、Display B を使用してスマートウォッチに心拍数を正しく入力できた。全体の平均としては目標心拍数より Display A では 1.8 回少なく、Display B では 1.6 回少ないという結果となった。

目標心拍数が大きくなるほど、スマートウォッチで取得される心拍数はマイナス方向への誤差が大きくなっている。これは処理時間の同期の精度の低さや、ディスプレイのリフレッシュレートが関係していると考えられる。図 1 のとおり、提案手法では $T[s]$ ごとに色を更新していく必要があるが、Python と Processing でデータのやり取りなどもする必要があり処理時間により遅延が生じてしまい、本来ディスプレイに 60 秒間で描画すべき回数よりも描画回数が少なくなってしまう可能性がある。この遅延時間は、処理回数の増加にともなって増加すると考えられる。そのため、目標心拍数が大きくなるほど処理回数が増加し、ディスプレイへの描画の遅延が増加してしまい右肩下りの傾向になったと考えられる。また、リフレッシュレートが低いと、正しく全ての色を $T[s]$ ごとに描画できない可能性がある。リフレッシュレートについては、ディスプレイの性能に依存する。

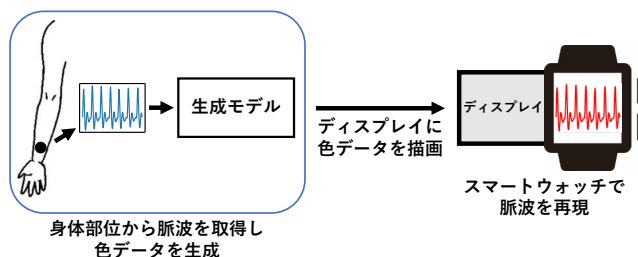


図 6 身体部位から得られた脈波データを入力して再現する機構

5. 今後

評価実験ではディスプレイを用いてスマートウォッチに 3 回以内の精度で、目標とする心拍数を計測させることができた。しかしながら、ディスプレイに対する制約が存在するほか、装着位置によっても取得される値がシビアに変化してしまう可能性が考えられる。今後は、実環境での使用を想定して脈波データの再現性を高めつつ、身体部位から得られた実際の脈波データを入力することで、ディスプレイ上に装着したウェアラブルデバイスに同一の脈波データを計測させる機構を実装する。この機構の想定を図 6 に示す。実現するには、ディスプレイに描画する色を自動で決定し続ける必要がある。そのため、脈波データを入力することでディスプレイに描画する色を出力することができるよう生成モデルを構築していく。また、身体部位に装着して使用することを想定し、デバイスの小型化を進めていくほか、複数のウェアラブルデバイスを使用して実験を行っていく。

6. まとめ

本研究では、ディスプレイを用いて光電脈波センサに任意の心拍数を計測させる手法を提案した。ディスプレイ描画プログラムおよび、スマートウォッチアプリケーションを実装し、スマートウォッチとディスプレイ 2 台を使用して、提案手法の効果を確認するための評価実験を行った。その結果、入力した目標心拍数と実際にスマートウォッチで取得された心拍数の誤差が全体で 3 回以内だった。ディスプレイごとの平均は Display A で -1.8 回、Display B で -1.6 回という結果であり、高い精度で心拍数が再現できた。

今後はディスプレイを用いることで、身体部位から取得された実際の脈波データと同様の脈波データをウェアラブルデバイスに計測させる機構を実装する。そのためには、自動でディスプレイの色調を決定していく必要があるため、適切な生成モデルを設計していく。また、身体部位への装着を想定し、デバイスの小型化を検討していく。

謝辞 本研究は、科学技術振興機構戦略的創造研究推進事業さきがけ (JPMJPR1937) の支援を受けたものである。ここに記して謝意を表す。

参考文献

- [1] Anastasiia Havriushenko, Kostyantyn Slyusarenko, and Illia Fedorin. Smartwatch based respiratory rate estimation during sleep using cnn/lstm neural network. In *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 584–587, 2020.
- [2] Jooyeun Ham, Jonggi Hong, Youngkyoon Jang, Seung Hwan Ko, and Woontack Woo. Smart wristband: Touch-and-motion-tracking wearable 3d input device for smart glasses. In *Distributed, Ambient, and Pervasive Interactions*, pp. 109–118, 2014.
- [3] J. Hernandez, Y. Li, J. M. Rehg, and R. W. Picard. Bioglass: Physiological parameter estimation using a head-mounted wearable device. In *2014 4th International Conference on Wireless Mobile Communication and Healthcare - Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MO-BIHEALTH)*, pp. 55–58, 2014.
- [4] A. Nishajith, J. Nivedha, S. S. Nair, and J. Mohammed Shaffi. Smart cap - wearable visual guidance system for blind. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 275–278, 2018.
- [5] A. Vahdatpour, N. Amini, and M. Sarrafzadeh. On-body device localization for health and medical monitoring applications. In *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 37–44, 2011.
- [6] Timo Sztyler, Heiner Stuckenschmidt, and Wolfgang Petrich. Position-aware activity recognition with wearable devices. *Pervasive and Mobile Computing*, Vol. 38, pp. 281 – 295, 2017.
- [7] Kai Kunze, Paul Lukowicz, Holger Junker, and Gerhard Tröster. Where am i: Recognizing on-body positions of wearable sensors. In *Location- and Context-Awareness*, pp. 264–275, 2005.
- [8] Kazuki Yoshida and Kazuya Murao. Estimating load positions of wearable devices based on difference in pulse wave arrival time. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC)*, p. 234–243, 2019.
- [9] S. Spinsante, S. Porfiri, and L. Scalise. Accuracy of heart rate measurements by a smartwatch in low intensity activities. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–6, 2019.
- [10] Sougata Sen, Vigneshwaran Subbaraju, Archan Misra, Rajesh Krishna Balan, and Youngki Lee. The case for smartwatch-based diet monitoring. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 585–590, 2015.
- [11] Andrew H. Johnston and Gary M. Weiss. Smartwatch-based biometric gait recognition. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–6, 2015.
- [12] Gary M. Weiss, Jessica L. Timko, Catherine M. Gallagher, Kenichi Yoneda, and Andrew J. Schreiber. Smartwatch-based activity recognition: A machine learning approach. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 426–429, 2016.
- [13] Dimitrios Iakovakis and Leontios Hadjileontiadis. Standing hypotension prediction based on smartwatch heart rate variability data: A novel approach. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, p. 1109–1112, 2016.
- [14] Taylor R. Mauldin, Marc E. Canby, Vangelis Metsis, Anne H. H. Ngu, and Coralys Cubero Rivera. Smartfall: A smartwatch-based fall detection system using deep learning. *Sensors*, Vol. 18, No. 10, 2018.
- [15] Lucio Ciabattini, Francesco Ferracuti, Sauro Longhi, Lucia Pepa, Luca Romeo, and Federica Verdini. Real-time mental stress detection based on smartwatch. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 110–111, 2017.
- [16] Dong Han, Syed Khairul Bashar, Fahimeh Mo-hagheghian, Eric Ding, Cody Whitcomb, David D. McManus, and Ki H. Chon. Premature atrial and ventricular contraction detection using photoplethysmographic data from a smartwatch. *Sensors (Basel, Switzerland)*, Vol. 20, No. 19, p. 5683, 2020.
- [17] Atefeh Goshvarpour, Ataollah Abbasi, and Ateke Goshvarpour. Fusion of heart rate variability and pulse rate variability for emotion recognition using lagged poincare plots. *Australasian Physical & Engineering Sciences in Medicine*, Vol. 40, No. 3, pp. 617–629, Sep 2017.
- [18] Yusuke Kajiwara, Toshihiko Shimauchi, and Haruhiko Kimura. Predicting emotion and engagement of workers in order picking based on behavior and pulse waves acquired by wearable devices. *Sensors*, Vol. 19, No. 1, 2019.
- [19] Min Seop Lee, Yun Kyu Lee, Dong Sung Pae, Myo Taeg Lim, Dong Won Kim, and Tae Koo Kang. Fast emotion recognition based on single pulse ppg signal with convolutional neural network. *Applied Sciences*, Vol. 9, No. 16, 2019.
- [20] Abdallah Ismail. What should my heart rate be? 03 2018.