

令和 5 年度

卒業研究報告書

仮想筋電義手の開発に関する研究

指導教官 戸崎哲也

報告者 河合将暉

神戸市立工業高等専門学校

電子工学科

(論文要旨)

ああ
ああ
ああ
ああ
ああ

目次

第1章:	はじめに	2
第2章:	解説	3
2.1	EinScan HX	3
2.1.1	製品仕様	3
2.2	Blender	3
2.2.1	スムージング	3
2.2.2	ボーン構成	4
2.2.3	ウェイトペイント	4
2.3	Unity	4
2.3.1	座標系	4
2.3.2	シェーダー	4
	既存シェーダー	4
	Reflex Shader 2.2	4
2.3.3	カメラ	4
2.3.4	C#スクリプト	4
2.3.5	オブジェクト	4
2.3.6	衝突判定	4
2.3.7	ペアレント	4
2.3.8	コンストレイント	4
2.3.9	物理演算	4
2.3.10	エディタ設定	4
2.3.11	プラグイン	4
	FVRsdk	4
	Xcharts	4
	Android Logcat	4
	XR Interaction Toolkit	4
	Google VR	4
2.3.12	ビルド	4
	iOS	5
	Android OS	5
2.4	FirstVR	5
2.4.1	デバイス構成	5
2.4.2	筋変位センサ	5
2.4.3	トラッキング	5
2.4.4	キャリブレーション	5
2.4.5	BLE 通信	5
2.5	Meta Quest2	5
2.5.1	デバイス構成	5
2.6	rasberry py	5
第3章:	研究手順	6
3.1	使用器具	6
3.2	3D スキャナ	6
3.2.1	3D モデルの取り込み	6
3.2.2	出力形式	6

3.3	Blender	6
3.3.1	スムージング処理	6
3.3.2	ボーン配置	7
3.4	Unity	7
3.4.1	オブジェクトの処理	7
3.4.2	ステージ構成	7
3.4.3	アニメーション設定	7
	アニメーションクリップ作成	7
	アニメーター設定	7
3.4.4	オブジェクト保持	7
3.4.5	衝突判定	7
3.4.6	腕の回転	7
3.4.7	カメラ位置	7
3.4.8	コンストレイント	7
3.4.9	UI	7
3.5	FirstVR	7
3.6	ビルド	7
3.7	評価	7
3.8	FirstVR	7
第4章:	研究結果	8
第5章:	まとめ	9
第6章:	今後の課題	10
	参考文献	12

第 1 章 はじめに

上肢切断者が筋電義手を装着する際、自在に扱うことができるように訓練を行う必要がある。VR を用いた筋電義手トレーニングの効果については先行研究 [1][2] で検討されているが、3D モデルのリアリティについて検討されていなかったため、本研究では仮想筋電義手モデルのリアリティによる訓練効果や幻肢痛緩和効果に着目し 3D スキャナで取り込んだ仮想筋電義手モデルを用いた VR トレーニングシステムを構成し、その見た目によって訓練効果に変化があるかを評価することを目的とする。また、近年では“カグラ”[6] という製品は上肢機能障害者のリハビリテーションのために運用されるなど VR トレーニングシステムは義手装具者以外にも需要が高まっている。

本研究は 2023 年度、神戸高専 Digital Fabrication Lab に新規導入されたハンディ 3D スキャナ“Ein Scan HX”[5] の活用方法の一例としても後続の研究に活用していただきたい。

第 2 章 解説

本章では，本研究で利用した“Ein Scan HX”をはじめ各種ソフトウェアの利用方法およびその使い方について解説していく．また，機器を選定した理由についても記述する．

2.1 EinScan HX

EinScan HX は株式会社サンステラが提供するハンディ 3D スキャナである．主な使用用途としては，工業製品などの比較的大きな物をスキャンし，リバースエンジニアリングや測定などに用いられている．製品仕様については次項で解説する．

2.1.1 製品仕様

EinScan HX の製品仕様について表 2.1 に示す．

表 2.1: EinScan HX の製品仕様

スキャン形式	Rapid スキャン	レーザースキャン
スキャン精度	0.05 mm	0.04 mm
ポイント間隔	0.25～3.00 mm	0.05～3.00 mm
被写体長 3D 精度	±0.1 mm	±0.06 mm
シングルスキャン精度	420 × 440 mm	
光源	ブルー LED	ブルーレーザー 7 本
被写体深度		
対象物との距離		
テクスチャスキャン		
安全性		
データ出力		
本体サイズ		
本体重量		

ここで，被写体長 3D 精度というのはスキャン時に取得するポイントの最大累積誤差を示したもので，測定する点数が増える場合や取得点の距離が大きい場合に誤差が累積し，大きくなっていく．

2.2 Blender

Blender とは，オープンソースの完全無料統合型 3DCG・2D・映像編集ソフトウェアである．本研究では“EinScan HX”によって出力した.obj 形式の 3D モデルを編集する目的で使用した．次項からは本研究で用いた Blender の機能について解説する．

2.2.1 スムージング

スムージング機能とは，3D オブジェクト表面をペイントソフトのようになぞるだけで表面を平滑化し，頂点を揃える機能である．

2.2.2 ボーン構成

ボーンは、3D オブジェクトを変形させる際に頂点の移動を制御する支柱の役割を果たす機能である。

2.2.3 ウェイトペイント

ウェイトペイントは、ボーンによって制御された頂点の変形の度合いをスペクトルで表示したもので、赤になるほど大きく変形し、青に近づくほど変形しなくなる。関節部分は赤色に、それ以外は青色にウェイトペイントを設定する。自然な変形のためには、関節部分の赤色からグラデーションのように広げていくと 3D オブジェクトが自然に変形する。

2.3 Unity

Unity は、

2.3.1 座標系

2.3.2 シェーダー

既存シェーダー

Reflex Shader 2.2

2.3.3 カメラ

2.3.4 C#スクリプト

2.3.5 オブジェクト

2.3.6 衝突判定

2.3.7 ペアレント

2.3.8 コンストレイント

2.3.9 物理演算

2.3.10 エディタ設定

2.3.11 プラグイン

FVRsdk

Xcharts

Android Logcat

XR Interaction Toolkit

Google VR

2.3.12 ビルド

本項では Unity で構成したシミュレータのビルド方法について解説する。以下にビルドの手順を示す。

1. Unity エディタ上のツールバーから”File”を選択
2. “Build Settings”の項目を選択すると、別ウィンドウに Build Settings が開かれる
3. 追加したいシーンファイルを Unity エディタ上で開いておき、Build Settings のウィンドウから”Add Open Scenes”を選択

4. Build Settings ウィンドウの左にある Platform から実行したい OS を選択し、ウィンドウ右下の”Switch Platform”を選択
5. Build Settings ウィンドウ右下の”build”を選択してビルド開始

以上の手順で Unity で作成したシーンのビルドが完了する。Windows,Linux などの OS ではビルドしたフォルダに実行ファイルが作成されているため、そのファイルからシミュレータを実行することができる。iOS,Android OS に関してはビルド・実行方法が異なるため、それぞれ以下に解説する。

iOS

本項では、iOS 用アプリの実行手順を以下に示す。

1. ビルドされたフォルダを MacOS に転送
2. MacOS でビルドされたフォルダを開き、フォルダ内の “Unity-iPhone.xcodeproj” を Xcode で開く
3. iOS 端末を接続し、端末の設定から端末をデバッグモードにする
4. Xcode ウィンドウ上部の Any iOS Device を選択し、接続した端末を選択
5. Signing & Capabilities を選択し、Team の欄に Apple ID を入力
6. 本シミュレータでは Bluetooth を用いるため、info を選択し、Key の一覧に “Privacy - Bluetooth Peripheral Usage Description” を追加し、Value を “Uses BLE to communicate with devices.” にする
7. iPhone にアプリがインストールされたら、端末の『設定』から『プライバシーとセキュリティ』を開き『Bluetooth』を選択
8. インストールしたアプリに Bluetooth の権限を許可する
9. アプリ一覧からインストールしたアプリを実行する

以上の手順で iOS 用のシミュレータをビルドして実行できる。この手順で一番重要なのが、6 の手順でこれがないとアプリを実行しても Made by Unity のポップがでたあと直後に動かなくなってしまう。これらのエラーは Xcode 上でログが残されているため、発生しているエラーを解消すれば実行することができる。一般的に、端末側の権限付与ができておらずアプリが実行できない場合があるためよく確認をすること。

Android OS

2.4 FirstVR

2.4.1 デバイス構成

2.4.2 筋変位センサ

2.4.3 トラッキング

2.4.4 キャリブレーション

2.4.5 BLE 通信

2.5 Meta Quest2

2.5.1 デバイス構成

2.6 raspberry py

第 3 章 研究手順

3.1 使用器具

本研究での使用器具を表 3.1 に示す。

表 3.1: 本研究における使用器具

No	機器名	型番	シリアル No	備考
1	EinScan HX			
2	FirstVR	UHL-01		
3	iPhone SE2			iOS16.7.2
4	HUAWEI Nova Lite2			AndroidOS 9
5	PC			Ubuntu22.04
6	PC			MacOS

以下に本研究で使用したソフトウェアを表 3.2 に示す。

表 3.2: 使用ソフトウェア

No	ソフトウェア名	バージョン	使用 OS	備考
1	Blender	3.0.1	Ubuntu	
2	Unity	2022.3.11f1	Ubuntu	
3	Xcode	15.0.1	MacOS	

3.2 3D スキャナ

3.2.1 3D モデルの取り込み

ハンディ 3D スキャナである EinScan HX を用いて左腕をスキャンした

3.2.2 出力形式

本研究では、腕の 3D オブジェクトにテクスチャを用いることを前提としているため、スキャン形式を Rapid スキャンモードで、3D モデルのリアリティをあげるために頂点数を 50 万点で出力し、出力形式は元からテクスチャがメッシュに割当されている.obj 形式にした。

3.3 Blender

3.3.1 スムージング処理

.obj 形式で取り込んだ 3D オブジェクトは測定によるノイズが含まれており、特に掌と手の甲の境界線上に段差のように途切れてしまう。このノイズを除去・補完するために、オブジェクト表面の凹凸を平坦にする効果があるスムージング処理を行った

3.3.2 ボーン配置

3.4 Unity

3.4.1 オブジェクトの処理

3.4.2 ステージ構成

3.4.3 アニメーション設定

アニメーションクリップ作成

アニメーター設定

3.4.4 オブジェクト保持

3.4.5 衝突判定

3.4.6 腕の回転

3.4.7 カメラ位置

3.4.8 コンストレイント

3.4.9 UI

3.5 FirstVR

3.6 ビルド

3.7 評価

3.8 FirstVR

本研究で構成するシミュレータの入力感度を検証するため、以下の手順で FirstVR の評価を行った。

1. 肘から手首までの長さを測定し、肘を原点に 1/4 の距離で装着
2. 端末と FirstVR を接続し、ジェスチャをしていない状態とジェスチャしている状態を学習
3. sample 数を 7,10～100 までの 11 段階で FirstVR の筋変位センサの出力値を測定
4. 最もノイズの少ない sample 数を選出し、ジェスチャ状態で手の角度を上下左右に動かした場合のノイズを測定

第 4 章 研究結果

第 5 章 まとめ

第 6 章 今後の課題

謝辭

参考文献

- [1] 芝軒 太郎 他.“VR を利用した筋電義手操作トレーニングシステムの開発と仮想 Box and Block Test の実現”. JRSJ. 2012 July.
- [2] Osumi M, et al. “Characteristics of Phantom Limb Pain Alleviated with Virtual Reality Rehabilitation”. Pain Med. 2019 May.
- [3] H2L.Inc.,Tokyo106-0032,Japan;satoshi.hosono@h2l.jp
- [4] Tamon Miyake, etal“Gait Phase Detection Based on Muscle Deformation with Static Standing-Based Calibration”. MDPI. 2021 Feb
- [5] 株式会社サンステラ, <https://www.einscan.jp/einscan-hx>
- [6] mediVR.Inc.,<https://www.medivr.jp/>