

2022 年度(令和四年度)
創造工学セミナーⅡ Final Report

骨格角度を用いた着座作業時の 姿勢評価方法の提案

研究メンバー

S519003 泉翔太

S519006 今成遥飛

S519012 小澤佑起

指導教員

金丸隆志 教授

所属研究室

知能機械研究室

目次

1	緒論 (今成)	4
1.1	近年の日本における着座時間の増加と健康的被害 (今成)	4
1.1.1	近年の日本における着座作業時間の増加	4
1.1.2	着座作業時間の増加による健康的な被害	6
1.1.3	姿勢と身体的症状の関係	8
1.2	姿勢評価研究の紹介 (今成)	9
1.2.1	参考研究「デスクワーク時における着座姿勢の評価方法の提案」	9
1.2.2	参考研究「デジタル傾斜計を用いた胸椎後弯角に関する 評価方法の検者内・検者間信頼性」	10
1.2.3	参考論文「姿勢の評価と治療アプローチ」	11
1.2.4	参考資料「X線画像読影に必要な計測法」	11
1.3	本研究の目的 (今成)	12
2	姿勢評価システムの概要 (今成、泉)	13
2.1	概要 (今成)	13
2.2	使用機器 (今成)	14
2.2.1	Azure Kinect	14
2.2.2	デスクトップ型 PC	15
2.3	撮影時の環境 (泉)	16
2.4	Azure Kinect を用いた姿勢評価 (泉)	17
2.4.1	骨格推定	18
2.4.2	角度算出手法	20
3	上半身の姿勢評価 (泉)	21
3.1	上半身における胸椎後弯角の推定 (泉)	21
3.2	点 C' の決定 (泉)	23
3.2.1	個人差について	25
3.3	胸椎後弯角 θ_{tk} の評価実験 (泉)	27
3.3.1	姿勢変化に伴った評価角度の変化	27
3.3.2	各姿勢の角度の t 検定	29
4	首部の姿勢評価 (泉)	30
4.1	首部における胸椎後弯角の推定 (泉)	30
4.2	点 E' の決定 (泉)	32
4.2.1	個人差について	34
4.3	頸椎前弯角 θ_{cl} の評価実験 (泉)	35
4.3.1	姿勢変化に伴った評価角度の変化	35

4.3.2	各姿勢の角度の t 検定	38
5	評価システムの開発 (小澤)	39
5.1	良姿勢の評価基準について (小澤)	39
5.2	評価システムの流れ (小澤)	42
5.2.1	座高測定モード	43
5.2.2	姿勢評価モード	44
5.3	有限オートマトン (FSM) について (小澤)	47
5.3.1	有限オートマトンを用いたモード切り替え	48
6	姿勢評価システムの評価実験 (小澤)	50
6.1	実験方法 (小澤)	50
6.2	結果 (小澤)	50
7	結論と課題 (小澤)	55
7.1	結論	55
7.2	課題	55
	参考文献	56
	謝辞	58
	付録	59

1 緒論 (今成)

本章では、近年の日本において生活様式の変化が健康的な問題を引き起こす一因となっていることについてまず述べる。そして、この問題を解決するために「理学療法的な知見」に注目し、〈簡易的な座位姿勢評価システム〉の開発を目標としたことを説明する。

1.1 近年の日本における着座時間の増加と健康的被害 (今成)

本節では、近年の日本において着座作業時間が増加していることをはじめに示し、それによって健康的な被害が増えていることを示す。

1.1.1 近年の日本における着座作業時間の増加

近年の日本では、新型コロナウイルスの影響もあり、〈テレワーク〉や〈オンライン授業〉といった〈PCを用いたオンライン形式の生活様式〉が普及している。例えば図 1 は、総務省が平成 24 年から令和 3 年にかけて行った「通信利用動向調査」[1]の結果をまとめたグラフであり、ここ数年における企業のテレワーク導入状況の推移を表している。令和元年から令和 2 年にかけての伸びが顕著であり、現在では過半数の企業がテレワークを導入していることがわかる。

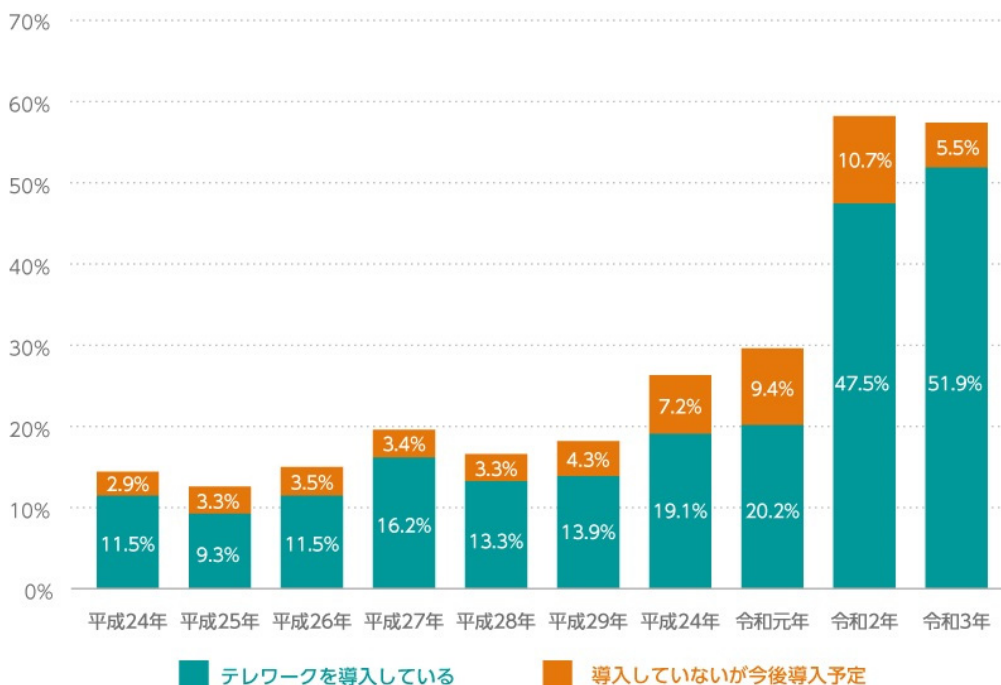


図 1 企業のテレワーク導入状況の推移[1]

また、図 2 は e ラーニング戦略研究所が 2020 年 7 月に発表した、オンライン授業の実施率に関する調査結果[2]である。この調査は 2020 年 6 月に、全国の大学職員・教員 100 名を対象に実施された。これによると、大学のオンライン授業の実施率は 97%であることがわかる。

したがって、大学教育現場におけるオンライン授業の導入も全国的に進んでいることがわかる。

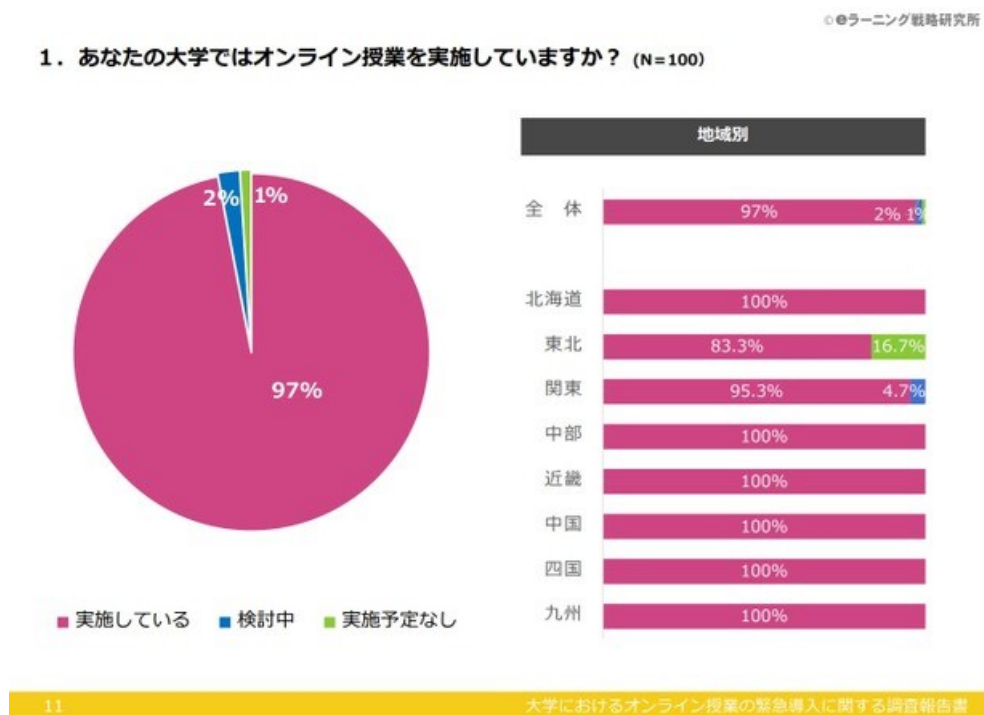


図 2 大学のオンライン授業の実施率[2]

このように、職場や大学において<テレワーク>や<オンライン授業>が普及することは、人々が椅子や床に座って作業を行う時間が増えることに繋がる。したがって、近年の日本では着座作業時間が増加傾向にあると考えられる。

1.1.2 着座作業時間の増加による健康的な被害

次に、1.1.1項で説明したことを受け、長時間座っての作業による健康被害について述べる。図3、図4はオムロンヘルスケア株式会社が発行した2020年度に20代から50代のテレワークをしている男女1,000人を対象に行った、「テレワークとなった働き世代への緊急アンケート」[3]の結果である。このアンケート結果によると、働き世代のうち約3割の人がテレワーク開始後に身体的な不調を感じていること、そして、不調の内訳として「肩こり」「精神的なストレス」「腰痛」といった症状が続くことがわかる。

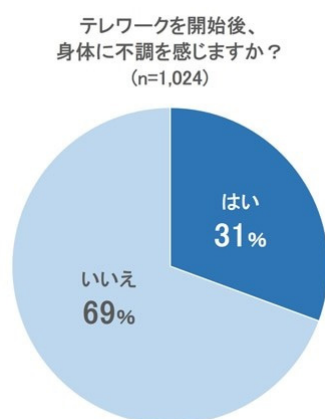


図3 テレワーク開始後の身体に不調を感じる割合[3]

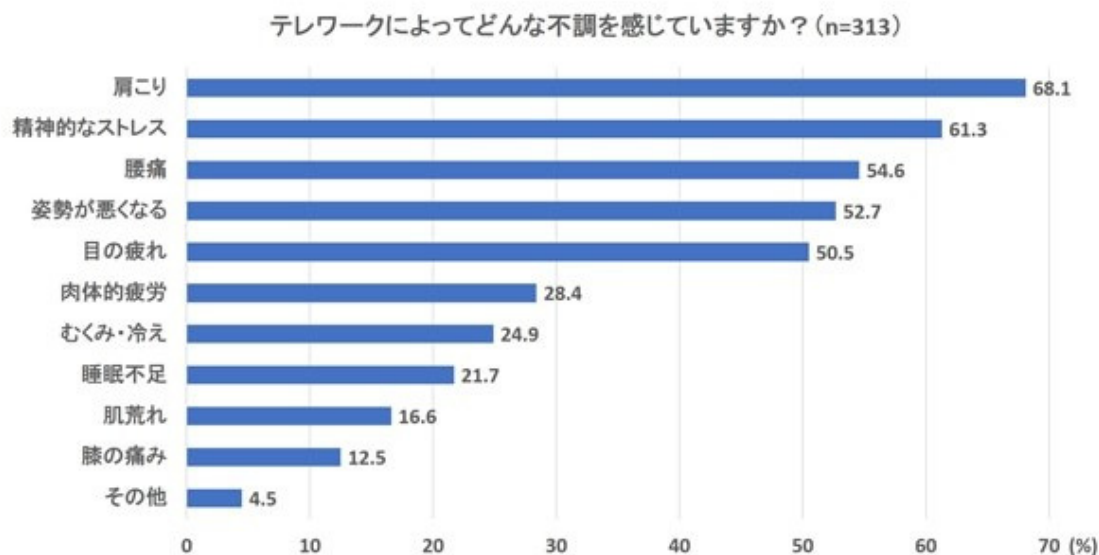


図4 不調を感じた人による不調の詳細[3]

また、表1は早稲田大学が2020年8月3日～22日にかけて実施した、「オンライン授業に関するアンケート調査」[4]の結果である。これによると、「身体的な疲れを感じる」と答えた学生が6割以上であることがわかる。

表 1 オンライン授業に関する調査結果[4]

オンライン授業の良い点		オンライン授業の改善点	
自宅で学習できる	76.4%	課題が多い	63.0%
自分のペースで学習できる	70.3%	目や耳、肩など身体的な疲れをより感じる	61.8%
通学時間を学習に有効活用できる	57.0%	友達と一緒に学べず孤立感を感じる	58.1%
復習が何度でもできる	48.4%	課題提出ができていないのか不安だった	52.9%

このように、<テレワーク>や<オンライン授業>をはじめとした<PC を用いたオンライン形式の生活様式>は、生活の利便性が向上する一方、「腰痛」や「肩こり」をはじめとした<身体的症状>を増加させる一因となっている。

1.1.3 姿勢と身体的症状の関係

1.1.2項に示した<健康的な被害>を引き起こす原因として、我々は作業時の姿勢に着目した。

高桑氏らによる高校生の肩こりに関するアンケート結果[5]によると、「他人から姿勢が悪いと日頃から指摘されると答えた生徒に肩こりが多いことも猫背などによる姿勢性の因子も肩こりの一因であることを示唆するものと考えられる。」という記述がある。

また、高井氏らによる課題遂行時の座位姿勢と心身の急性ストレス反応に関する研究[6]では、一時的に「姿勢を正す」ことが、適切な生理的反応の誘発と回復を導くことや、生体内に留まらず主観的反応の自覚にまで至ることを示しており、姿勢が心身の両方に作用することを示唆している。

このように、姿勢と身体的症状には密接な関係があり、悪い姿勢が身体に悪影響を及ぼすと考えられる。さらに、日常から良い姿勢を意識することや、時折姿勢を正すことは心身の健康に繋がることがわかる。これらより、我々は座位姿勢に着目し、悪い姿勢であればそれを評価するシステムの開発が望まれると考えた。

1.2 姿勢評価研究の紹介 (今成)

本節では、本研究で参考にした研究を紹介する。我々は、1.2.1項に示すような着座作業時の姿勢評価システムに、1.2.2項～1.2.4項に示す理学療法的な知見を組み合わせることにより良い評価システムの開発が可能になると考えた。

1.2.1 参考研究「デスクワーク時における着座姿勢の評価方法の提案」

本項では、内藤氏らが行った着座姿勢の評価方法に関する研究[7]を紹介する。この研究では、姿勢検知の精度を高めるために、距離と体圧に関する2種類のセンサーを用いる評価方法の提案をしている(図5)[7]。また、座り姿勢を独自で定めた点数方式で評価しており、評価が低い際にユーザーをリラックスさせるための「リコメンデーション」を発生させる方式をとっている。

我々は、姿勢をリアルタイムで分析する点、姿勢が悪くなった際にリコメンデーションを表示する点の2点を参考にした。また、我々はより簡易的なシステムを目指し、カメラを用いて人物の特徴が調べられないかを検討する。



図 5 デスクワーク時における着座姿勢の評価方法の提案[7]

1.2.2 参考研究「デジタル傾斜計を用いた胸椎後弯角に関する 評価方法の検者内・検者間信頼性」

本項では、大村氏らによる、デジタル傾斜計を用いた胸椎後弯角の評価方法に関する研究[8]について紹介する。胸椎後弯とは 12 対の椎体で構成される生理学的な弯曲のことを言う。

図 6 は胸椎後弯角の測定方法に関する図である。測定方法は「解剖学的立位姿勢を基準とし、第 1 胸椎棘突起（以下、Th1）に傾斜計の上端を、第 12 胸椎棘突起（以下、Th12）に傾斜計の下端を合わせ、床面からの垂直線に対する Th1 の前傾角度と Th12 の後傾角度を測定した。Th1 および Th12 の測定した値を加算したものを胸椎後弯角とした」と述べられている（図 6）。また、この研究での健常者の胸椎後弯角の測定結果を表 2 に示す。この方法では測定に際し、デジタル傾斜計と触診の技術が必要となることが問題点として挙げられている。

我々はデジタル傾斜計を使用して胸椎後弯角を測定する手法を参考にする。

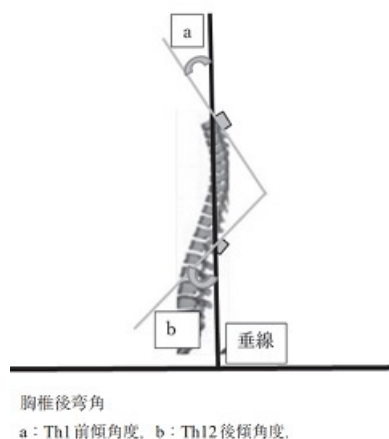


図 6 デジタル傾斜計を用いた胸椎後弯角の測定[8]

表 2 健常者の胸椎後弯角の測定結果[8]

検者	1 回目	2 回目
A	34.4 ± 1.9	35.0 ± 1.6
B	35.7 ± 2.0	36.6 ± 2.0

単位：°, n=10, 平均値 ± 標準偏差.

1.2.3 参考論文「姿勢の評価と治療アプローチ」

姿勢評価に関連し、竹井氏が筋・筋膜を中心とした姿勢の評価と治療アプローチについて概説した論文[9]を紹介する。図 7 は、立位における身体各部の正常なアライメントを表した図であり、頸椎前弯角は約 30～35 度、胸椎後弯角は約 40 度、腰椎前弯角は約 45 度が正常であるとしている。

我々は、理学療法士の姿勢評価の観点と、身体各部の正常なアライメントにおける生理的湾曲の角度の 2 点を参考にした。

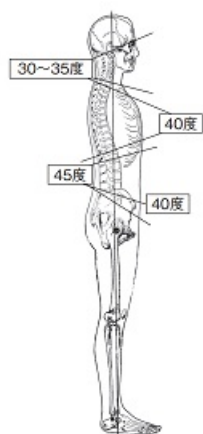


Fig.1 矢状面の正常な立位姿勢

図 7 立位における身体各部の正常なアライメント[9]

1.2.4 参考資料「X線画像読影に必要な計測法」

本項では、市川氏によって解説された、X線画像読影のためのX線計測法と判定基準に関する資料[10]を紹介する。特に、角度を測定する頸椎側面による前弯・後弯の評価法に関して、「頸椎 C1 椎体の上縁線と C7 椎体下縁線とのなす角度を測定する。」と述べられている。

そこで、我々は頸椎前弯角の測定方法として参考にした。また、この測定方法では、評価の際にX線による大掛かりな撮影装置が必要になることが問題として挙げられた。

1.2.3 項と 1.2.4 項から、我々は理学療法を用いた頸椎前弯角の測定方法を参考にする。

1.3 本研究の目的 (今成)

1.1 節～1.2 節より、我々は以下の条件を満たす〈姿勢を簡易的に測定できるシステム〉を開発することを目的とした。

1. カメラを用いて人物の特徴を認識できること
2. 定量的な姿勢評価を行えること
3. リアルタイムに姿勢を分析し、悪い姿勢であれば改善策を表示できること

1 の条件を満たすために、我々は Microsoft 社製の Azure Kinect DK (以下 Azure Kinect) を使用することとした。Azure Kinect は、画像処理により人体の骨格を取得できるセンサーである。詳細は 2.2.1 項に示す。

2 の条件を満たすために、PC によって数値的に姿勢を分析する手法を確立する。

3 の条件を満たすために、良い姿勢の定義とそれに伴い姿勢状態によって改善点をアドバイスするためのユーザーインターフェース (以下、ガイダンス UI) を作成する。

2 姿勢評価システムの概要 (今成、泉)

2.1 概要 (今成)

姿勢評価システムの利用イメージ図を図 8、フローチャートを図 9 に示す。システムの入力はユーザーの動作であり、出力は分析結果から得られた情報を PC モニターに表示したものである。

まず、Azure Kinect によって取得した映像から人物の骨格推定を行い、その骨格点から姿勢評価に必要な角度を算出する。その結果に応じて、PC モニター上に姿勢ガイダンスを表示する。この一連の流れはリアルタイムで更新される。

また、Azure Kinect を用いた姿勢評価の詳細は 3 章と 4 章で述べる。

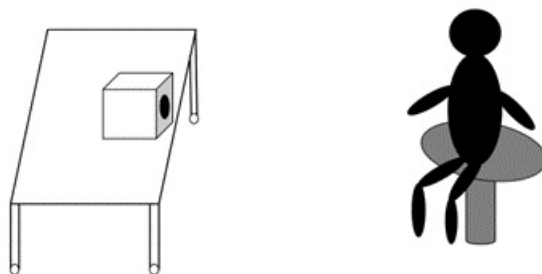


図 8 システムの概要

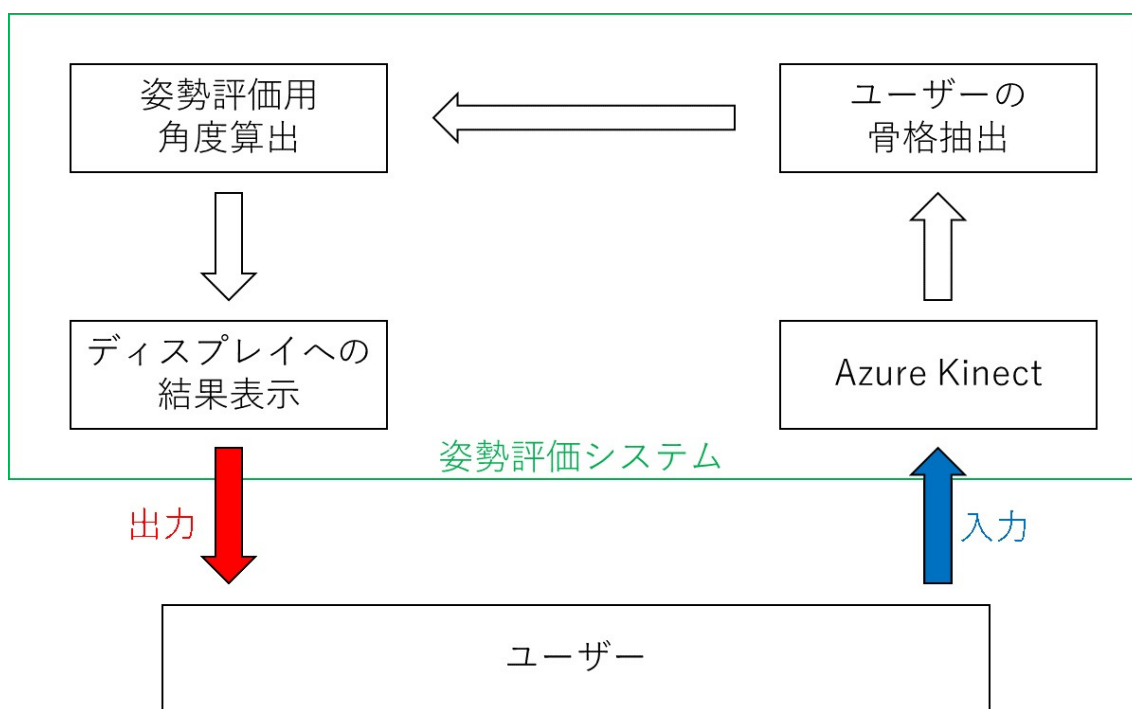


図 9 システムのフローチャート

2.2 使用機器 (今成)

本節では、本システムで使用した機器である Azure Kinect およびデスクトップ型 PC についての詳細を述べる。

2.2.1 Azure Kinect

我々は、骨格推定用のハードウェアとして Azure Kinect (図 10) を使用する。Azure Kinect で取得できるカラー映像の解像度は 3840×2160 pixel である。また、人物の検出範囲は 0.25 ~ 2.88m である。詳しい仕様を表 3 に示す。Azure Kinect を PC で利用するためには SDK である Azure Kinect SDK および Azure Kinect Body Tracking SDK[11]が必要である。



図 10 Azure Kinect

表 3 Azure Kinect の仕様

RGB カメラ	解像度	3940×2160
	FPS	30
深度カメラ	解像度	512×512
	FPS	30
関節		32 関節/人
人物の検出範囲		0.25~2.88m
映像が取得可能な角度	水平	120 度
	垂直	120 度
IMU		3 軸加速度 + 3 軸角速度
接続性	通信	USB3.1Gen1/w type C
	電源	AC アダプターまたは USB C
筐体	サイズ	$103 \times 39 \times 126$ mm
	重量	440g

2.2.2 デスクトップ型 PC

図 11 に本研究で使したデスクトップ型 PC、表 4 にその仕様を示す。



図 11 使したデスクトップ型パソコン

表 4 デスクトップ型 PC の仕様

OS	Windows10
CPU	Intel® CORE™ i3-8100
RAM	8.0GB
GPU	NVIDIA GeForce GTX 1070

2.3 撮影時の環境 (泉)

本研究では 1 章で述べた研究目的と背景を踏まえ、椅子への着座状態での姿勢評価に焦点を当てている。本節では、その状態を計測する環境について解説する。

図 12 のように、Azure Kinect と被験者の距離を 1900mm とし、床からの Azure Kinect の高さを 700mm として撮影を行う。また、撮影方向については、背中や首の曲がり具合を目で見て判断できるように、横からの撮影とする。

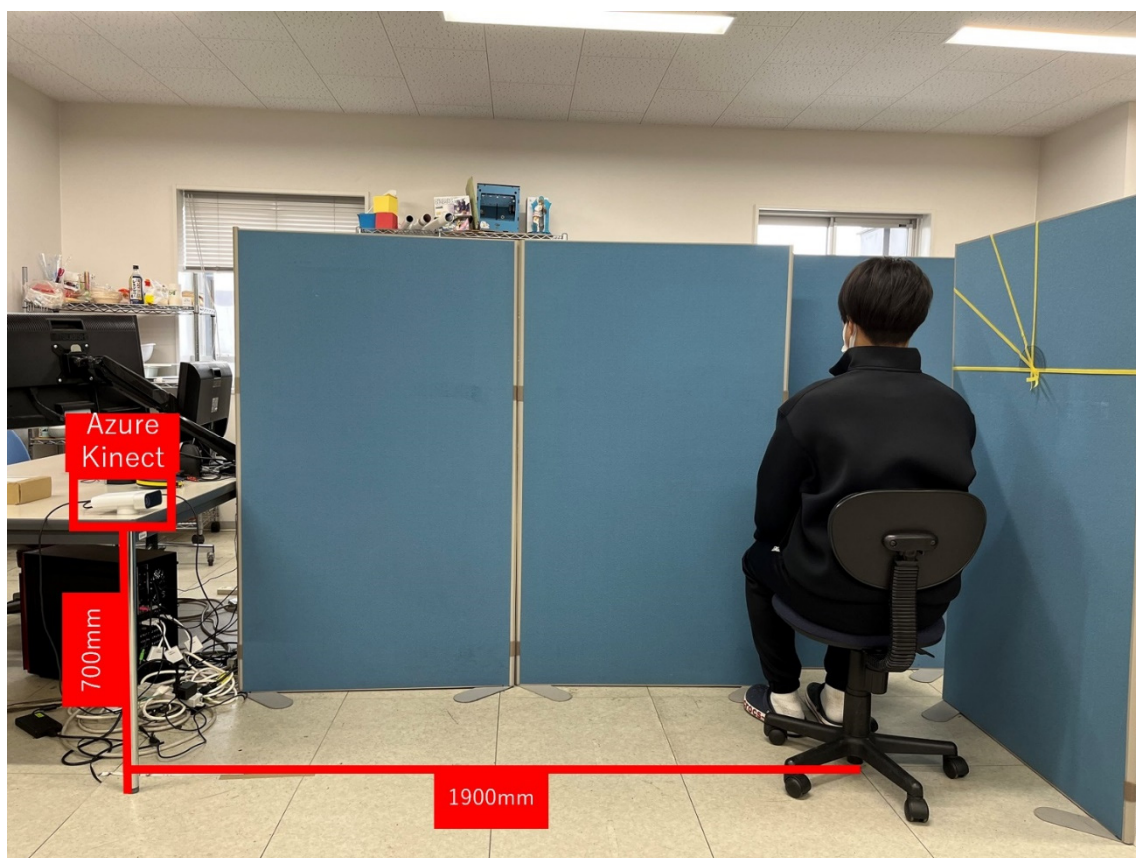


図 12 撮影環境

2.4 Azure Kinect を用いた姿勢評価（泉）

1 章で述べた研究目的を踏まえ、我々は Azure Kinect を用いた姿勢評価方法の提案を行う。本システムは以下の 2 つから構成される。

- 骨格推定
- 角度算出手法

骨格推定技術を用いることで、人間の姿勢状況を目視よりも正確に捉えることが可能になる。詳細については 2.4.1 項で述べる。

角度算出手法を用いることで、姿勢を数値化して扱うことが可能になる。詳細については 3 章と 4 章で述べる。

これら 2 つの結果をもとに姿勢評価を行う。

2.4.1 骨格推定

撮影された人物の骨格推定には、Azure Kinect の Body Tracking SDK を使用する。Body Tracking SDK により取得できる 32 個 (No. 0~No. 31) の骨格点の名称を表 5 に、その位置を図 13 に示す[12]。図 14 は実際に骨格点を取得した様子である。

表 5 Body Tracking SDK により得られる骨格点の名称

0	PELVIS	16	HANDTIP_RIGHT
1	SPINE_NAVAL	17	THUMB_RIGHT
2	SPINE_CHEST	18	HIP_LEFT
3	NECK	19	KNEE_LEFT
4	CLAVICLE_LEFT	20	ANKLE_LEFT
5	SHOULDER_LEFT	21	FOOT_LEFT
6	ELBOW_LEFT	22	HIP_RIGHT
7	WRIST_LEFT	23	KNEE_RIGHT
8	HAND_LEFT	24	ANKLE_RIGHT
9	HANDTIP_LEFT	25	FOOT_RIGHT
10	THUMB_LEFT	26	HEAD
11	CLAVICLE_RIGHT	27	NOSE
12	SHOULDER_RIGHT	28	EYE_LEFT
13	ELBOW_RIGHT	29	EAR_LEFT
14	WRIST_RIGHT	30	EYE_RIGHT
15	HAND_RIGHT	31	EAR_RIGHT

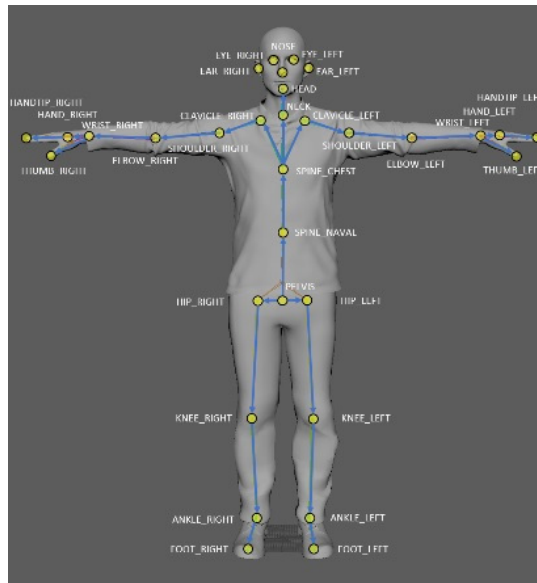


図 13 Body Tracking SDK により得られる骨格点の位置:[12]

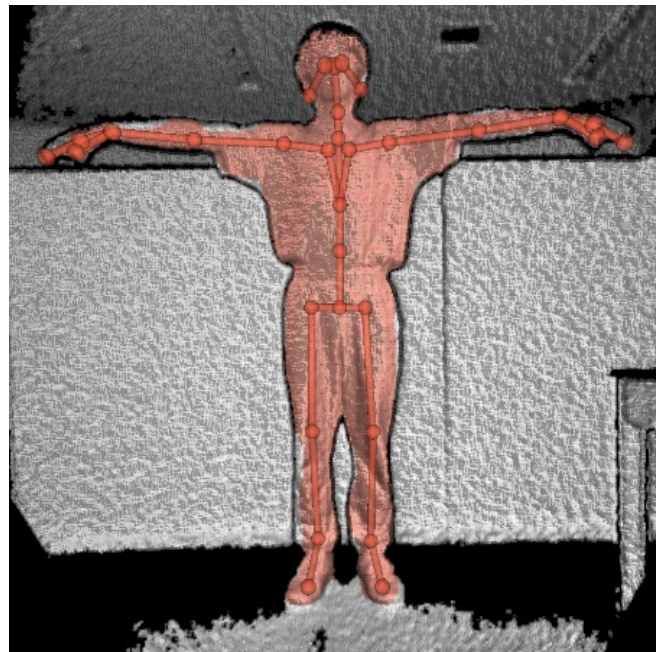


図 14 実際に骨格点を取得して表示した様子

2.4.2 角度算出手法

骨格推定により、得られたデータから姿勢を評価するために、何らかの角度を用いる。例えば、図 15 のように、着目する 3 点の内角などである。しかし、Azure Kinect の骨格点は解剖学的な骨格とは無関係であるため、理学療法的知見に基づいた角度とはならない。そのため、何らかの方法で Azure Kinect の骨格点と解剖学的な骨格を対応づける必要がある。その手法を 3 章と 4 章で解説する。

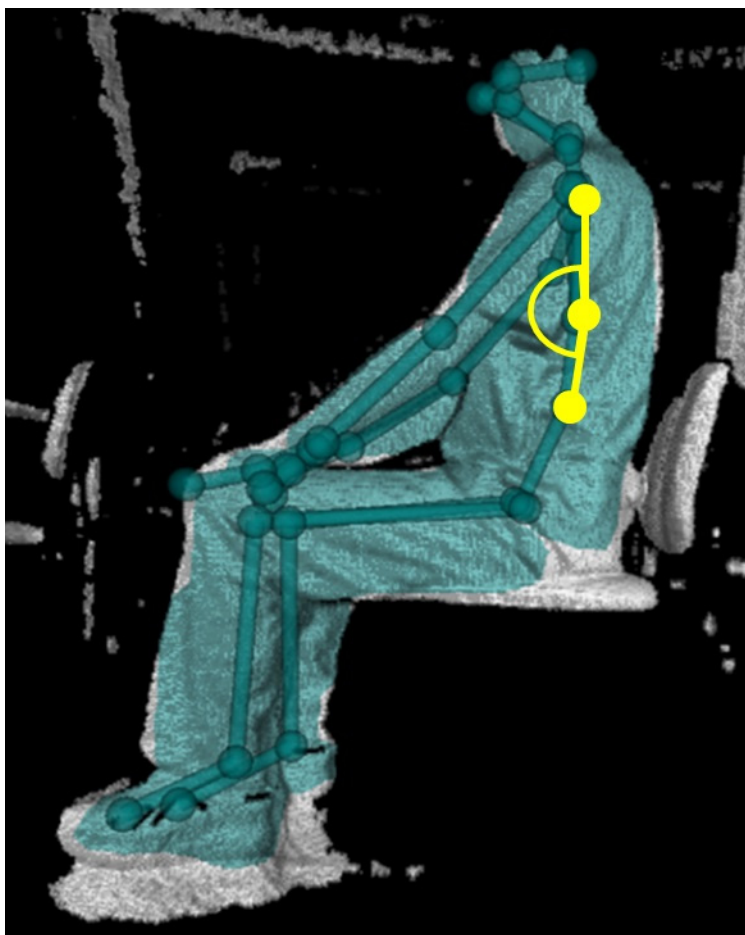


図 15 角度評価手法イメージ図

3 上半身の姿勢評価 (泉)

本章では、上半身について Azure Kinect の骨格点と解剖学的な骨格を対応づけ、姿勢評価に用いる角度を求める手法を提案する。

3.1 上半身における胸椎後弯角の推定 (泉)

上半身の評価では 1.2.2 項で述べた胸椎後弯角を用いる。本研究では胸椎後弯角 (Thoracic Kyphosis Angle) を θ_{tk} と表記する (図 16)。

θ_{tk} の計算には図 16 のように第 1 胸椎棘突起 (Th1) と第 12 胸椎棘突起 (Th12) が必要である。Azure Kinect の骨格点は推定骨格点であり、Th1 部と Th12 部に該当する骨格点が存在しない。疑似的に図 16 に近い状況を作り出すため、Azure Kinect の骨格点以外に人体解剖図 (図 17) 上に Th1 と Th12 に相当する新たに骨格点を定義する必要がある。

我々が考案した Th1 は図 18 では点 A とされており、この点は Azure Kinect の点 HEAD と SPINE_CHEST 間の線分を 2:3 に内分する点と定義した。また、Th12 は図 18 では点 B とされており、この点は Azure Kinect の点 SPINE_CHEST と SPINE_NAVAL 間の線分を 2:1 に内分する点と定義した。点 C は点 A と点 B を 37:46 に内分する点とし、点 C' は点 C から AB に対して垂直に伸ばした点とした。以上の比率は人体解剖図 (図 17) をもとに類推した。

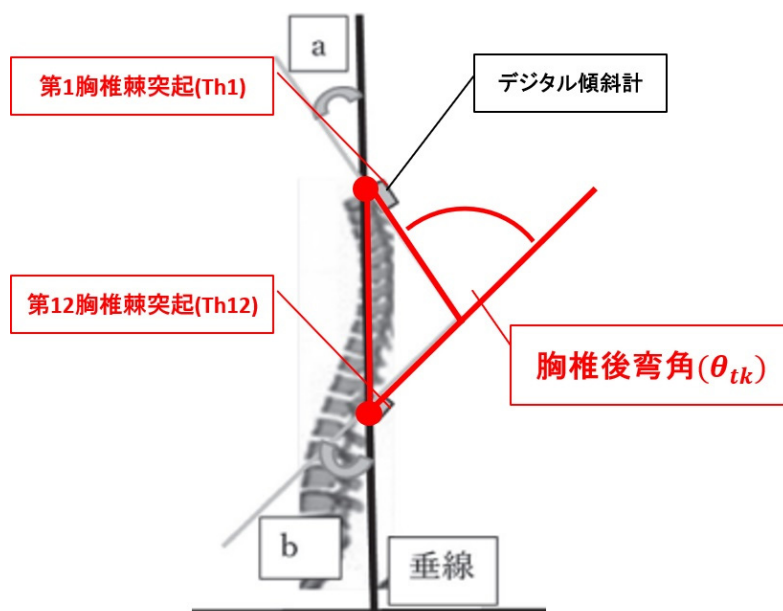


図 16 胸椎後弯角 θ_{tk}

(a : Th1 前傾角度, b : Th2 後傾角度)



図 17 人体解剖図

図 18 は参考手法の状況を Azure Kinect 上で再現した時のイメージ図である。
 点 C' が定まると仮定した場合、 θ_{tk} の算出方法は以下の式となる。

$$\theta_1 = \tan^{-1} \left(\frac{|\overline{CA}|}{|\overline{CC'}|} \right) \quad (1)$$

$$\theta_2 = \tan^{-1} \left(\frac{|\overline{CB}|}{|\overline{CC'}|} \right) \quad (2)$$

$$\theta_{tk} = 180 - (\theta_1 + \theta_2) \quad (3)$$

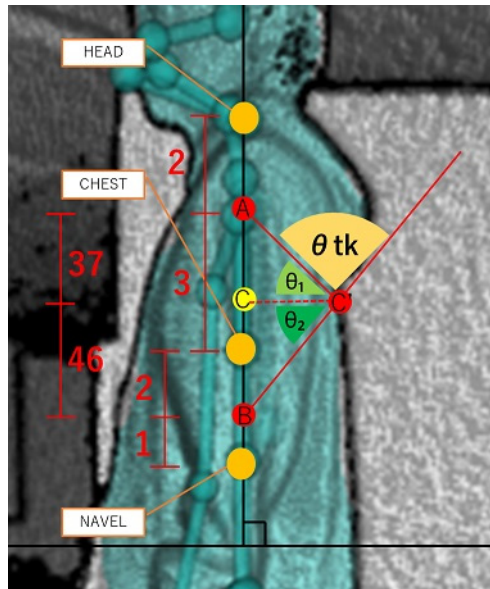


図 18 Azure Kinect 上の θ_{tk}

しかし、ここまででは点 C' の位置が定まっていないので θ_{tk} を計算することはできない。
 そこで、点 C' の決定方法の詳細を次節で説明する。

3.2 点C'の決定 (泉)

本節では図 18 の点 C' の決定方法について解説する。図 18 に示された通り、点 C' は脊柱上の点ではない。そのため、点 A、B、C とは異なり点 C' を図形的な手法では決定できない。本研究では、文献[8]における θ_{tk} の健常者の測定結果である $32.5^\circ \leq \theta_{tk} \leq 38.6^\circ \cdots \textcircled{1}$ を参考にする。 θ_{tk} が $\textcircled{1}$ の範囲に収まるよう、点 C' を決定する。手順は以下の通りである。

なお、 $|\overline{CC'}|$ (仮) は $\textcircled{1}$ の範囲を調査するために用いる未確定の $|\overline{CC'}|$ を指し、MIN は $\textcircled{1}$ の範囲に収まる $|\overline{CC'}|$ (仮) の下限値、MAX は $\textcircled{1}$ の範囲に収まる $|\overline{CC'}|$ (仮) の上限値を指す。

- $\textcircled{1}$ の範囲内となるように $|\overline{CC'}|$ (仮) を 1[mm]間隔でずらしながら、 $|\overline{CC'}|$ (仮) の範囲 ($MIN \leq |\overline{CC'}|$ (仮) $\leq MAX$) を求める。
- $|\overline{CC'}| = \frac{MAX+MIN}{2}$ とする。
- 求められた $|\overline{CC'}|$ を基に点 C' を決定する。

例： $|\overline{CC'}|$ (仮) の MIN が 37、MAX が 44 となった被験者を例として扱う。

$$37 \leq |\overline{CC'}| \text{ (仮)} \leq 44 \text{ となる場合、} |\overline{CC'}| = \frac{37+44}{2} = 40.5 \text{ [mm] とする。}$$

よって、この被験者の点 C' は点 C から AB に対して 40.5mm 垂直に離れた点となる。

このように、我々は健常者の θ_{tk} を満たす点 C' を作図により求めたが、 θ_{tk} から $|\overline{CC'}|$ を数学的に求める手法もある。以下にその方法を記す。

加法定理により

$$\tan(\theta_1 + \theta_2) = \frac{\tan \theta_1 + \tan \theta_2}{1 - \tan \theta_1 \tan \theta_2}$$

が得られるので、式(1)、(2)、(3)および $\tan(180 - \theta_{tk}) = -\tan \theta_{tk}$ を代入し整理すると、 $|\overline{CC'}|$ に関する以下の二次方程式が得られる。

$$\tan \theta_{tk} |\overline{CC'}|^2 + (|\overline{CA}| + |\overline{CB}|)|\overline{CC'}| - |\overline{CA}||\overline{CB}| \tan \theta_{tk} = 0$$

これを解くことで、以下のように $|\overline{CC'}|$ が θ_{tk} から求められる。

$$|\vec{CC'}| = \frac{-(|\vec{CA}| + |\vec{CB}|) + \sqrt{(|\vec{CA}| + |\vec{CB}|)^2 + 4|\vec{CA}||\vec{CB}|\tan^2 \theta_{tk}}}{2 \tan \theta_{tk}}$$

3.2.1 個人差について

我々の手法は比率を用いて骨格点 A、B を定義していることから、体格の個人差が変化することで $|\overline{CC'}|$ にも影響があると考えられる。そこで、研究室の 20 代前半男性 9 人の異なる体格での $|\overline{CC'}|$ を調査した。本実験では上半身と首部のように上半身のみの姿勢を評価するため、座高 (Z) を体格差の指標として用いる。座高の長さは表 6 の各骨格点間の距離を合計した値とする。

表 6 座高算出に用いた骨格点

骨格点番号	骨格点名
0	PELVIS
1	PSPINE_NAVAL
2	SPINE_CHEST
3	NECK
26	HEAD

PELVIS=P、SPINE_NAVAL=SN、SPINE_CHEST=SC、NECK=N、HEAD=H

$$l_{P\sim SN} = \sqrt{(P_x - SN_x)^2 + (P_y - SN_y)^2 + (P_z - SN_z)^2} \quad (4)$$

$$l_{SN\sim SC} = \sqrt{(SN_x - SC_x)^2 + (SN_y - SC_y)^2 + (SN_z - SC_z)^2} \quad (5)$$

$$l_{SC\sim N} = \sqrt{(SC_x - N_x)^2 + (SC_y - N_y)^2 + (SC_z - N_z)^2} \quad (6)$$

$$l_{N\sim H} = \sqrt{(N_x - H_x)^2 + (N_y - H_y)^2 + (N_z - H_z)^2} \quad (7)$$

$$\text{座高 } Z = (l_{P\sim SN} + l_{SN\sim SC} + l_{SC\sim N} + l_{N\sim H}) \quad (8)$$

図 20 は座高と $|\overline{CC'}|$ の関係を示した図である。図 20 からわかるように座高 (Z) の長さによって $|\overline{CC'}|$ が変化する。そこで座高 (Z) の長さによって $|\overline{CC'}|$ を変更する。以下の式 (9) が座高 (Z) によって変わる $|\overline{CC'}|$ を算出する式である。

$$|\overline{CC'}| \quad y = 0.0726x - 1.5365 \quad (9)$$

式 (9) の x に式 (8) で得た座高 (Z) の長さを代入することで、体格の個人差を反映した CC' の長さを決定することが出来る。

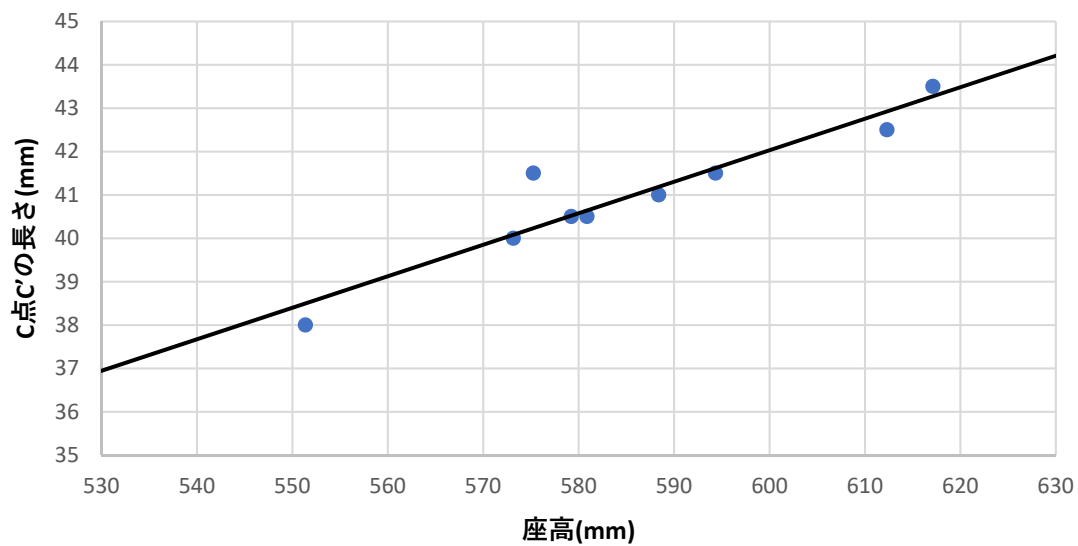


図 19 座高と CC' の長さの関係

3.3 胸椎後弯角 θ_{tk} の評価実験 (泉)

本節では 3.1 節で提案した胸椎後弯角 θ_{tk} が上半身の姿勢評価に適しているかを検討する。上半身において、姿勢変化時の θ_{tk} の変化についての詳細を 3.3.1 項で、各姿勢($0^\circ, 30^\circ, 60^\circ, 90^\circ$)を比較することが妥当であるかについての詳細を 3.3.2 項で述べる。

3.3.1 姿勢変化に伴った評価角度の変化

図 20 のように床から 90cm の位置 (上半身実験) にマスキングテープで $0^\circ, 30^\circ, 60^\circ, 90^\circ$ ごとに印をつけた壁を用意した。床に対して垂直な線を 0° とする。



図 20 上半身における実験環境

測定方法

被験者は 20 代前半男性 3 名とした。 $0^\circ, 30^\circ, 60^\circ, 90^\circ$ のテープに姿勢を合わせた状態 (図 21、図 22、図 23、図 24) をそれぞれの角度で、10 回ずつ記録を行う。再現性を確保するため、それぞれの角度のテープに頭頂部を合わせ、一度姿勢を記録したら立って歩くなどしてから次の記録に移行することとする。



図 21 0°のテープに姿勢を合わせた状態



図 22 30°のテープに姿勢を合わせた状態



図 23 60°のテープに姿勢を合わせた状態



図 24 90°のテープに姿勢を合わせた状態

測定結果

図 25 は各姿勢における θ_{tk} の変化を示している。誤差範囲は 30 データの標準偏差である。上半身を前に倒すにつれて、 θ_{tk} が大きくなることがわかる。

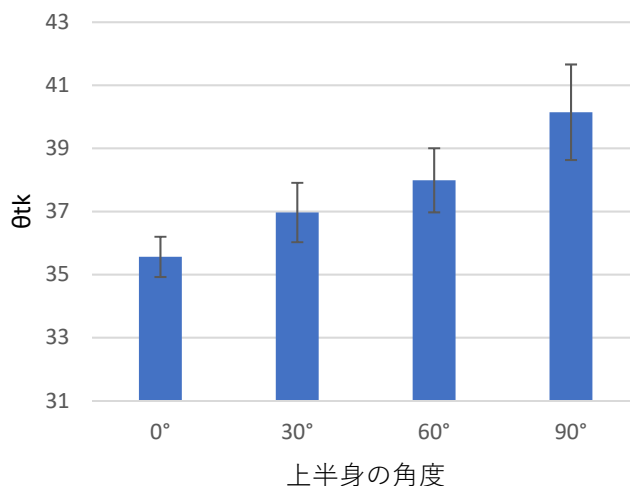


図 25 上半身における θ_{tk}

3.3.2 各姿勢の角度の t 検定

t 検定を用いて各姿勢で θ_{tk} が有意に変化したかを調べる。有意水準 5%未満で有意差ありとした検定結果を表 7 に示す。

表 7 上半身の角度変化の伴った θ_{tk} の変化

	0°	30°	60°	90°
θ_{tk}	35.6 ± 0.6° * ²⁾ * ³⁾ * ⁴⁾	37.0 ± 0.9° * ¹⁾ * ³⁾ * ⁴⁾	38.0 ± 1.0° * ¹⁾ * ²⁾ * ⁴⁾	40.1 ± 1.5° * ¹⁾ * ²⁾ * ³⁾

¹⁾ 0° と比較して有意差あり、²⁾ 30° と比較して有意差あり、³⁾ 60° と比較して有意差あり、⁴⁾ 90° と比較して有意差あり

*: p < 0.05

すべての角度間で有意差が見られたことから θ_{tk} を姿勢評価に利用できると考えられる。

4 首部の姿勢評価 (泉)

本章では、首部について Azure Kinect の骨格点と解剖学的な骨格と対応づけ、姿勢評価に用いる角度を求める手法を提案する。

4.1 首部における胸椎後弯角の推定 (泉)

首部では 1.2.3 項と 1.2.4 項で述べた頸椎前弯角を用いる。本研究では、頸椎前弯角 (Cervical Lordosis Angle) を θ_{cl} と表記する (図 26)。

θ_{cl} の計算には図 26 のように、第 1 頸椎 (C1) と第 7 頸椎 (C7) が必要である。Azure Kinect には第 1 頸椎に近い HEAD の骨格点は存在するが、第 7 頸椎に該当する骨格点が存在しない。そのため、上半身と同様の流れで疑似的に骨格点を定義する必要がある。

第 7 頸椎は胸椎 (Th1) と頸椎 (第 7 頸椎) の境界部であるため、上半身で定義した点 A と同じであるとみなした。そのため、上半身の時と同様に、Azure Kinect の骨格点 HEAD と SPINE_CHEST 間の線分を 2:3 に内分する点と定義した。第 1 頸椎は図 27 では点 D とされており、Azure Kinect の骨格点 HEAD と定義した。点 E は点 D と点 A を 7:29 に内分する点とし、点 E' は点 E から AD に対して垂直に伸ばした点とした。以上の比率は、人体解剖図 (図 17) をもとに類推した。

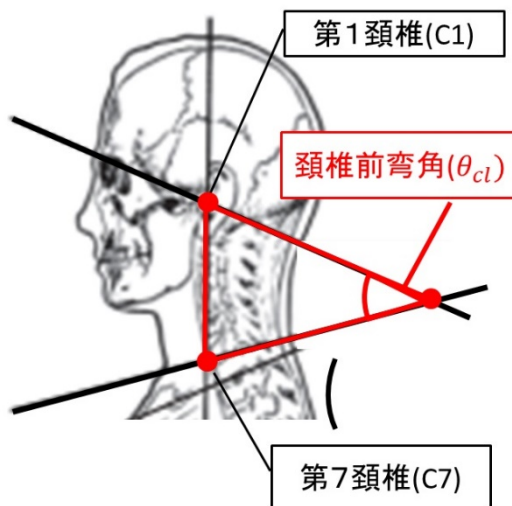


図 26 頸椎前弯角 θ_{cl}

図 27 は参考手法の状況を Azure Kinect 上で再現した時のイメージ図である。

点 E' が定まったと仮定した場合、 θ_{cl} の算出方法は以下の式となる。

$$\theta_3 = \tan^{-1} \left(\frac{|\overline{ED}|}{|\overline{EE'}|} \right) \quad (10)$$

$$\theta_4 = \tan^{-1} \left(\frac{|\overline{EA}|}{|\overline{EE'}|} \right) \quad (11)$$

$$\theta_{cl} = \theta_3 + \theta_4 \quad (12)$$

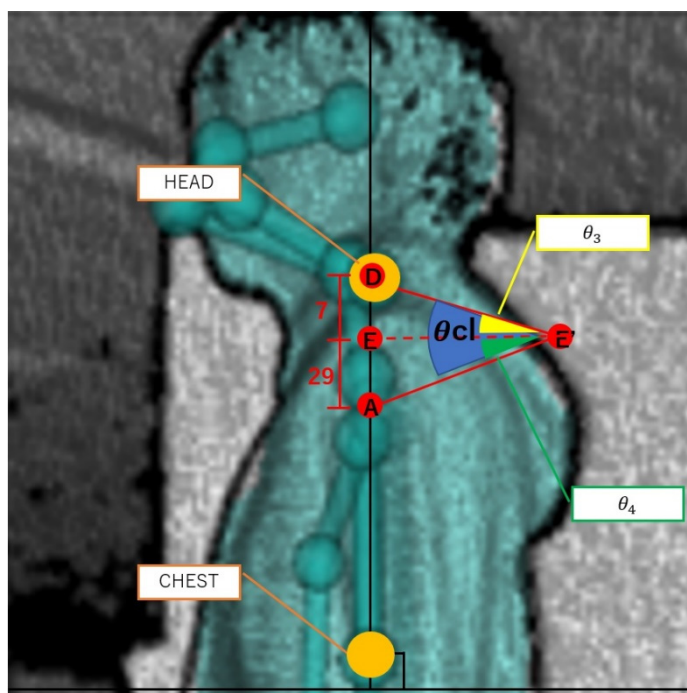


図 27 Azure Kinect 上の θ_{cl}

しかし、ここまでは点 E' の位置が定まっていないので θ_{cl} を計算することはできない。そこで、点 E' の決定方法の詳細を次節で説明する。

4.2 点 E' の決定 (泉)

本節では図 27 の点 E' の決定方法について解説する。図 27 に示された通り、点 E' は脊柱上の点ではない。そのため、点 A、D、E とは異なり点 E' を図形的な手法では決定できない。本研究では、文献[9]における θ_{cl} の正常値である $30^\circ \leq \theta_{cl} \leq 35^\circ$ …②を参考にする。

θ_{cl} が②の範囲に収まるよう、点 E' を決定する。手順は以下の通りである。なお、 $|\overline{EE'}|$ (仮)

は②の範囲を調査するために用いる未確定の $|\overline{EE'}|$ を指し、MIN は②の範囲に収まる $|\overline{EE'}|$ (仮)

の下限值、MAX は②の範囲に収まる $|\overline{EE'}|$ (仮) の上限値を指す。

- ②の範囲内となるように $|\overline{EE'}|$ (仮) を 1[mm]間隔でずらしながら、 $|\overline{EE'}|$ (仮) の範囲

($MIN \leq |\overline{EE'}|$ (仮) $\leq MAX$) を求める。

- $|\overline{EE'}| = \frac{MAX+MIN}{2}$ とする。

- 求められた $|\overline{EE'}|$ を基に点 E' を決定する。

例： $|\overline{EE'}|$ (仮) の MIN が 171、MAX が 202 となった被験者を例として扱う。

$$171 \leq |\overline{EE'}| \text{ (仮)} \leq 202 \text{ となる場合、} |\overline{EE'}| = \frac{171+202}{2} = 186.5[\text{mm}] \text{ とする。}$$

よって、この被験者の点 E' は点 E から AD に対して 186.5mm 垂直に離れた点となる。

このように、我々は健常者の θ_{cl} を満たす点 E' を作図により求めたが、 θ_{cl} から $|\overline{EE'}|$ を数学的に求める手法もある。以下にその方法を記す。

加法定理により

$$\tan(\theta_3 + \theta_4) = \frac{\tan \theta_3 + \tan \theta_4}{1 - \tan \theta_3 \tan \theta_4}$$

が得られるので、式(10)、(11)、(12)を代入し整理すると、 $|\overline{EE'}|$ に関する以下の二次方程式が得られる。

$$\tan \theta_{cl} |\overline{EE'}|^2 - (|\overline{ED}| + |\overline{EA}|) |\overline{EE'}| - |\overline{ED}| |\overline{EA}| \tan \theta_{cl} = 0$$

これを解くことで、以下のように $|\overrightarrow{EE'}|$ が θ_{cl} から求められる。

$$|\overrightarrow{EE'}| = \frac{(|\overrightarrow{ED}| + |\overrightarrow{EA}|) + \sqrt{(|\overrightarrow{ED}| + |\overrightarrow{EA}|)^2 + 4|\overrightarrow{ED}||\overrightarrow{EA}|\tan^2 \theta_{cl}}}{2 \tan \theta_{cl}}$$

4.2.1 個人差について

我々の手法は比率を用いて骨格点を定義していることから、体格の個人差が変化することによって $|\overline{EE'}|$ にも影響があると考えられる。そこで、研究室の20代前半男性9人の異なる体格での $|\overline{EE'}|$ を調査した。上半身と同様に、式(8)の座高(Z)を体格差の指標とする。

図28は座高と $|\overline{EE'}|$ の関係を示した図である。図28からわかるように座高(Z)の長さによって $|\overline{EE'}|$ が変化する。そこで、座高(Z)の長さによって $|\overline{EE'}|$ を変更する。以下の式(13)が座高によって変わる $|\overline{EE'}|$ を算出する式である。

$$|\overline{EE'}| \quad y = 0.3272x - 3.0457 \quad (13)$$

式(13)のxに式(8)で得た座高(Z)の長さを代入することで、体格の個人差に関係なく、 EE' の長さを求めることが出来る。

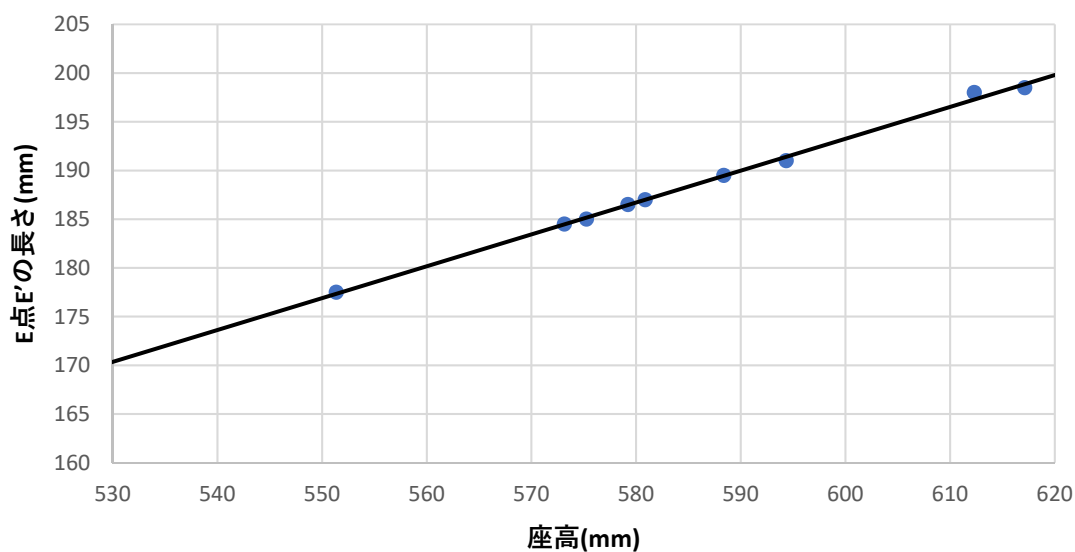


図 28 座高と EE' の長さの関係

4.3 頸椎前弯角 θ_{cl} の評価実験 (泉)

本節では 4.1 節で提案した頸椎前弯角 θ_{cl} が首部の姿勢評価に適しているかを検討する。首部において、姿勢変化時の θ_{cl} の変化についての詳細を 4.3.1 項で、各姿勢($0^\circ, 30^\circ, 60^\circ, 90^\circ$)を比較することが妥当であるかについての詳細を 4.3.2 項で述べる。

4.3.1 姿勢変化に伴った評価角度の変化

図 29 のように床から 110cm の位置 (首部実験) にマスキングテープで $0^\circ, 30^\circ, 60^\circ, 90^\circ$ ごとに印をつけた壁を用意した。床に対して垂直な線を 0° とする。

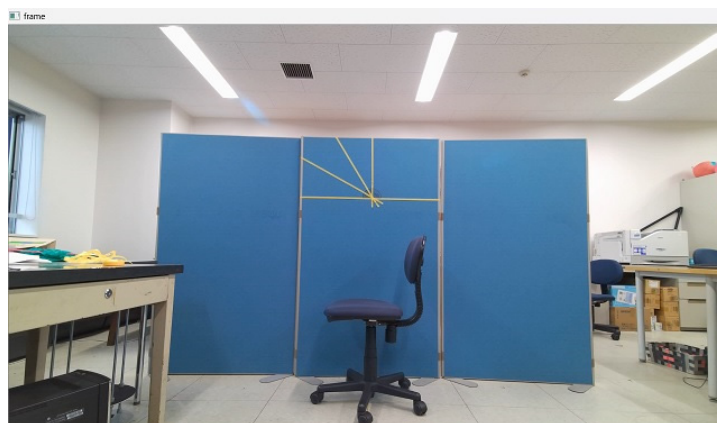


図 29 首部における実験環境

測定方法

被験者は 20 代前半男性 3 名で $0^\circ, 30^\circ, 60^\circ, 90^\circ$ のテープに姿勢を合わせた状態 (図 30、図 31、図 32、図 33) をそれぞれの角度で、10 回ずつ記録を行う。再現性を確保するため、それぞれの角度のテープに頭頂部を合わせ、一度姿勢を記録したら立って歩くなどしてから次の記録に移行することとする。



図 30 0°のテープに姿勢を合わせた状態



図 31 30°のテープに姿勢を合わせた状態



図 32 60°のテープに姿勢を合わせた状態



図 33 90°のテープに姿勢を合わせた状態

測定結果

図 34 は各姿勢における θ_{cl} の変化を示している。誤差範囲は 30 データの標準偏差である。首を前に倒すにつれて θ_{cl} が小さくなることがわかる。

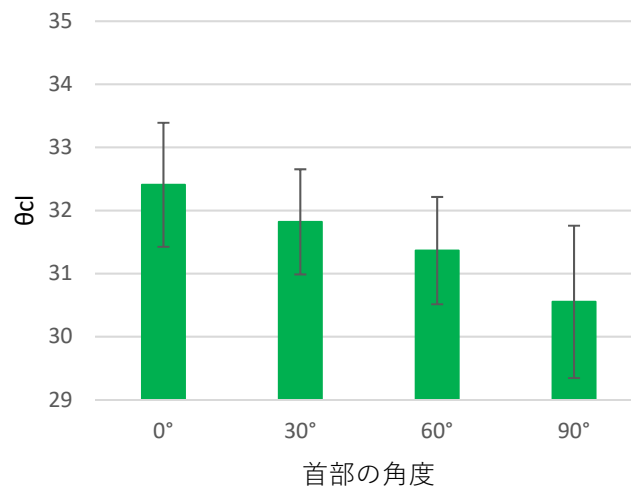


図 34 首節における θ_{cl}

4.3.2 各姿勢の角度の t 検定

t 検定を用いて各姿勢で θ_{cl} が有意に変化したかを調べる。有意水準 5%未満で有意差ありとした検定結果を表 8 に示す。

表 8 首部の角度変化の伴った θ_{cl} の変化

	0°	30°	60°	90°
θ_{cl}	32.4 ± 1.0° * ²⁾ ³⁾ ⁴⁾	31.8 ± 0.9° * ¹⁾ ³⁾ ⁴⁾	31.4 ± 1.0° * ¹⁾ ²⁾ ⁴⁾	30.6 ± 1.2° * ¹⁾ ²⁾ ³⁾

¹⁾0° と比較して有意差あり、²⁾30° と比較して有意差あり、³⁾60° と比較して有意差あり、

⁴⁾90° と比較して有意差あり

*:p<0.05

すべての角度間で有意差が見られたことから θ_{cl} を姿勢評価に利用できると考えられる。

5 評価システムの開発 (小澤)

本章では、評価システムの流れ (5.2 節) と有限オートマトンを用いたシステムのモード切り替え手法 (5.3 節) について説明する。

5.1 良姿勢の評価基準について (小澤)

文献[8]および[9]において、上半身の θ_{tk} と首部の θ_{cl} の正常値はそれぞれ $32.5^\circ \leq \theta_{tk} \leq 38.6^\circ$ (図 35 オレンジ枠)、 $30^\circ \leq \theta_{cl} \leq 35^\circ$ (図 36 オレンジ枠) とされている。我々の手法は理学療法による手法と完全には同じではないため、参考手法の範囲をそのまま用いることはできない。そのまま上記を評価基準値として用いてしまうと、ほとんどの姿勢が良姿勢と判定されてしまうからである。そこで、我々は上半身・首部ともに図 25 および図 34 における、 0° の誤差範囲を新たな良姿勢の評価基準値として用いることとする。

上半身 0° については、我々の測定した θ_{tk} は $35.6^\circ \pm 0.6^\circ$ であった。そこで、図 37 の赤枠 $35.0^\circ \leq \theta_{tk} \leq 36.2^\circ$ のときに上半身は良姿勢であると判定する。

首部 0° については、我々の測定した θ_{cl} は $32.4^\circ \pm 1.0^\circ$ であった。そこで、図 38 の赤枠 $31.4^\circ \leq \theta_{cl} \leq 33.4^\circ$ のときに上半身は良姿勢であると判定する。

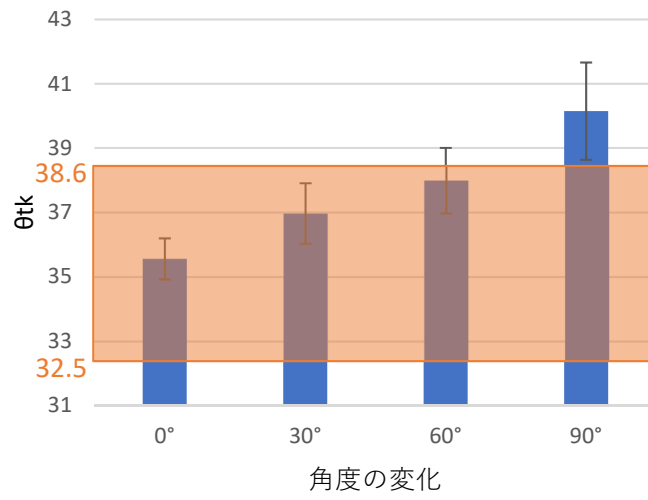


図 35 文献[8]における上半身の正常な角度の範囲

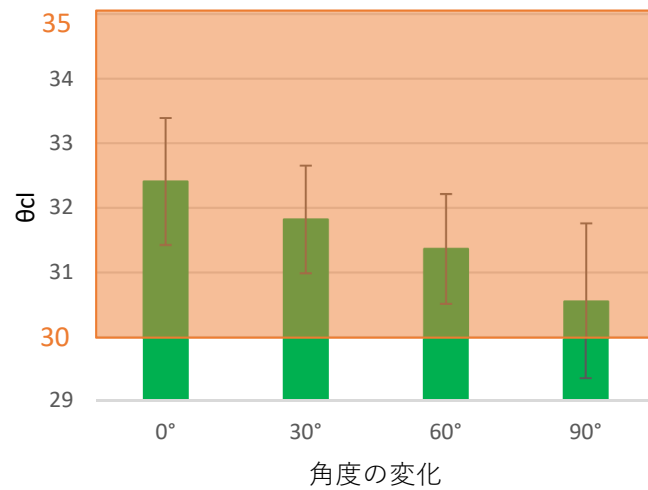


図 36 文献[9]における首部の正常な角度の範囲

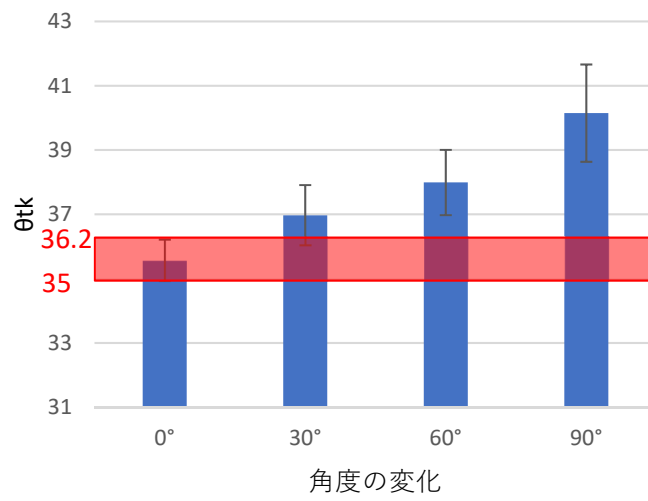


図 37 上半身における良姿勢の評価基準範囲

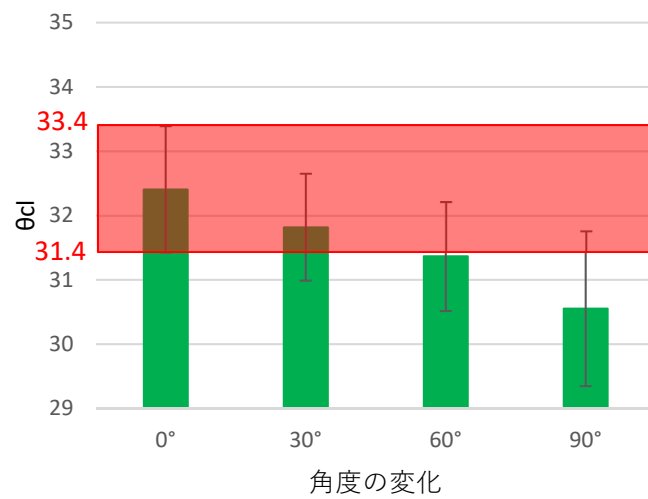


図 38 首元における良姿勢の評価基準範囲

5.2 評価システムの流れ (小澤)

Azure Kinect では、姿勢の違いや骨格推定の結果により座高の長さが常に変動する。この問題を解決するために、我々は良姿勢における座高を固定値として扱うことが必要だと考えた。

そこで、我々はシステムに対し、座高測定モード (5.2.1 節で説明する) と姿勢評価モード (5.2.2 節で説明する) という 2 つのモードを定義することにした。図 39 に示すように、座高測定モードにより被験者の座高を決定してから、姿勢評価モードに移行して評価を行う。

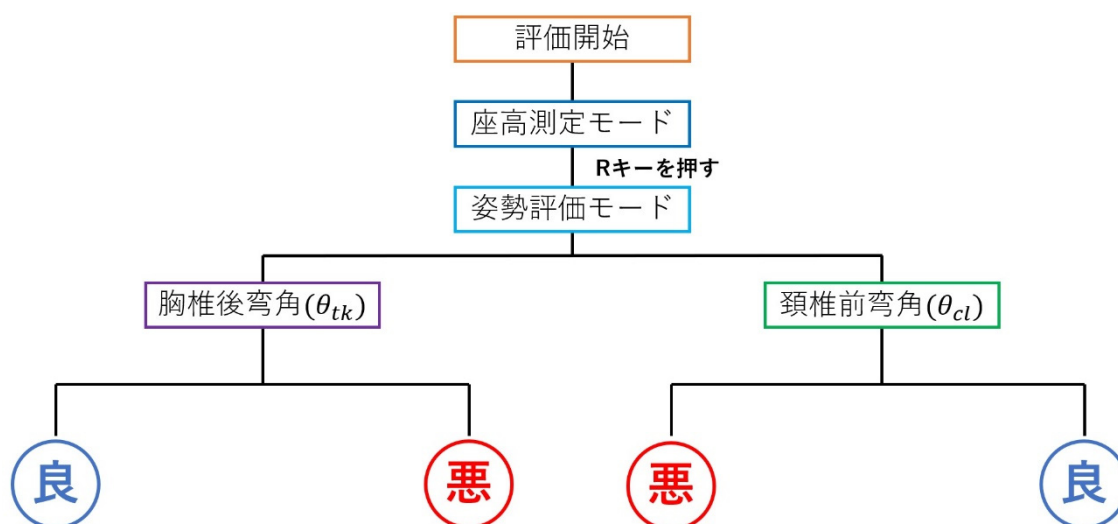


図 39 評価システムの流れ

5.2.1 座高測定モード

座高測定モードは、3.2.1項で座高として定義した座高Zの長さを Azure Kinect でリアルタイムに測定するモードである。座高はコード 1 より計算される。

```
#head から neck
HN = math.sqrt((head_x - neck_x)**2 + (head_y - neck_y)**2 + (head_z - neck_z)**2)
#neck から chest
NC = math.sqrt((neck_x - chest_x)**2 + (neck_y - chest_y)**2 + (neck_z - chest_z)**2)
#chest から naval
CN = math.sqrt((chest_x - naval_x)**2 + (chest_y - naval_y)**2 + (chest_z - naval_z)**2)
#naval から pelvis
NP = math.sqrt((naval_x - pelvis_x)**2 + (naval_y - pelvis_y)**2 + (naval_z - pelvis_z)**2)
#座高 z
z = (HN + NC + CN + NP)
```

コード 1 座高の計算を行うプログラム

5.2.2 姿勢評価モード

姿勢評価モードは、算出した θ_{tk} と θ_{cl} をもとに、姿勢評価を表示するモードである。 θ_{tk} と θ_{cl} はコード 2 のように算出する。また、姿勢ガイダンスを表示するプログラムはコード 3 である。

姿勢ガイダンスで表示される画像が図 40、図 41、図 42 である。コード 3 では、図 40 は `imggood`、図 41 は `img1`、図 42 は `img2` と定義されている。

これらを用いて、5.1 で述べた評価基準値の 2 つをもとに、姿勢ガイダンスを表示する表示する。



図 40 良姿勢を示す姿勢ガイダンス

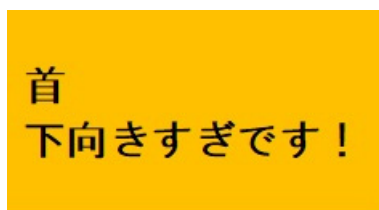


図 41 首において前傾姿勢を示す姿勢ガイダンス

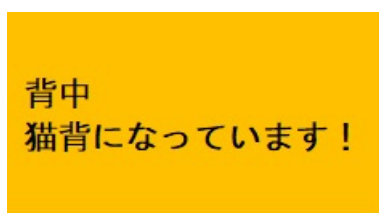


図 42 上半身において前傾姿勢を示す姿勢ガイダンス

```
# 上半身
a_x = (3 * head_x + 2 * chest_x) / 5
a_y = (3 * head_y + 2 * chest_y) / 5
a_z = (3 * head_z + 2 * chest_z) / 5

b_x = (2 * naval_x + 1 * chest_x) / 3
```

```

b_y = (2 * naval_y + 1 * chest_y) / 3
b_z = (2 * naval_z + 1 * chest_z) / 3

c_x = (37 * b_x + 46 * a_x) / 83
c_y = (37 * b_y + 46 * a_y) / 83
c_z = (37 * b_z + 46 * a_z) / 83

C_x = (CC/(math.sqrt((a_z - b_z)**2 + (b_x - a_x)**2)))*(a_z - b_z) + c_x
C_y = c_y
C_z = (CC/(math.sqrt((a_z - b_z)**2 + (b_x - a_x)**2)))*(b_x - a_x) + c_z

bx = b_x - C_x
by = b_y - C_y
bz = b_z - C_z

ax = a_x - C_x
ay = a_y - C_y
az = a_z - C_z

cos =
(ax*bx+ay*by+az*bz)/(math.sqrt(ax*ax+ay*ay+az*az)*math.sqrt(bx*bx+by*by+bz*bz)
)

ac = math.sqrt((a_x - c_x)**2 + (a_y - c_y)**2 + (a_z - c_z)**2)
bc = math.sqrt((b_x - c_x)**2 + (b_y - c_y)**2 + (b_z - c_z)**2)
alpha = math.atan(ac/CC)
beta = math.atan(bc/CC)
thetalrad = alpha + beta

thetal = math.degrees(thetalrad)
theta_TK = 180 - thetal

#頸椎
d_x = head_x
d_y = head_y
d_z = head_z

```

```

a_x = (3 * head_x + 2 * chest_x) / 5
a_y = (3 * head_y + 2 * chest_y) / 5
a_z = (3 * head_z + 2 * chest_z) / 5

e_x = (7 * a_x + 29 * d_x) / 36
e_y = (7 * a_y + 29 * d_y) / 36
e_z = (7 * a_z + 29 * d_z) / 36

de = math.sqrt((d_x - e_x)**2 + (d_y - e_y)**2 + (d_z - e_z)**2)
ae = math.sqrt((a_x - e_x)**2 + (a_y - e_y)**2 + (a_z - e_z)**2)
alpha = math.atan(de/EE)
beta = math.atan(ae/EE)
theta2rad = alpha + beta
theta_CL = math.degrees(theta2rad)

```

コード 2 θ_{tk} と θ_{cl} を算出するプログラム

```

img1 = cv2.imread('b1.png')
img2 = cv2.imread('c1.png')
imggood = cv2.imread('good.png')
height,width = img1.shape[:2]

#頸椎
if 31.4 < theta_CL < 33.4:
    frame[0:height,0:width] = imggood
else:
    frame[0:height,0:width] = img1

#胸椎
if 35 < theta_TK < 36.2:
    frame[200:height + 200,0:width] = imggood
else:
    frame[200:height + 200,0:width] = img2

```

コード 3 姿勢ガイダンスを表示するプログラム

5.3 有限オートマトン (FSM) について (小澤)

本研究では、5.2 節で述べたように、姿勢変化によって座高 (Z) が変わってしまう問題がある。座高 (Z) を固定値として扱うために、座高測定モードと姿勢評価モードの 2 つのモードを定義し、そのモードを切り替えている。そのモードの切り替えに、有限オートマトン (Finite State Machine, FSM) [13, 14] を用いる。

FSM とは、状態や遷移の数が有限個で表されるオートマトン (automaton: 自動人形を意味し、コンピュータの状態と遷移を表や図でモデル化したもの) である。ある状態から特定の条件で、別の状態に遷移する。なお、1 つの遷移条件に対して 1 つ状態が存在する。ある状態を初期状態と呼び、そこからオートマトンが始動するものである。また、遷移先の状態を最終状態あるいは受理状態と呼ぶ。

例えば、図 43 はじゃんけんにおいて相手がグーを出した時の有限オートマトンである。相手がグーを出した状態を初期状態とし、受理状態を勝ち、あいこ、負けの三つの状態とする。自分が出す手の形(グー、チョキ、パー)の入力によってそれぞれの状態に遷移する。

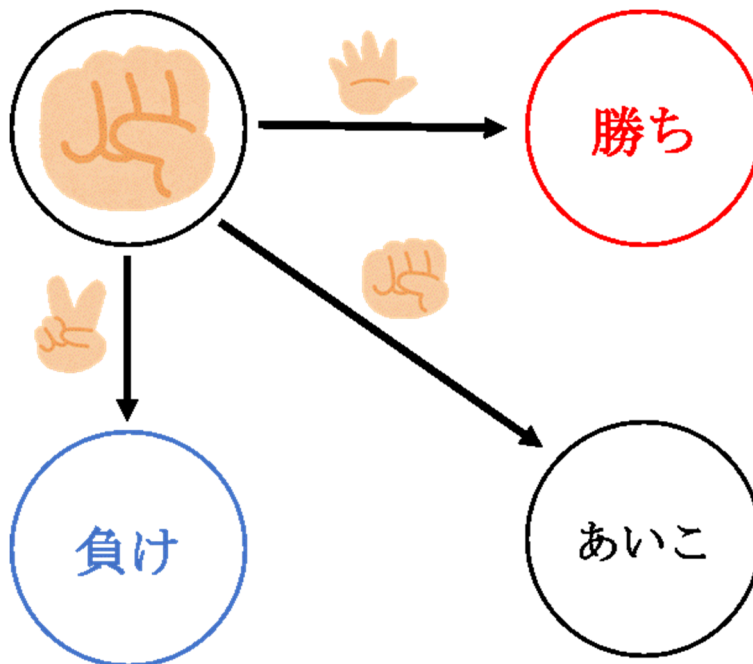


図 43 有限オートマトンのイメージ図[14]

本研究における FSM の利用の詳細を 5.3.1 項で説明する。

5.3.1 有限オートマトンを用いたモード切り替え

図 44 は、本システムの状態遷移図を示したものである。座高測定モードと姿勢評価モードの 2 つ状態および遷移条件を記している。

本システムでは、座高測定モードを初期状態（コード 4 では STATE_INITIALIZING）とし、姿勢評価モードを遷移後の状態（コード 4 では STATE_MEASURING）とする。遷移は、キーボードの R キーが押されたタイミングとする。

これらにより、システムが起動した後、座高(Z)を取得し、R キー押下時は姿勢評価モードに移行することを実現する。なお、この図からわかるように、本システムでは姿勢評価モードから座高測定モードへの遷移は想定していない。

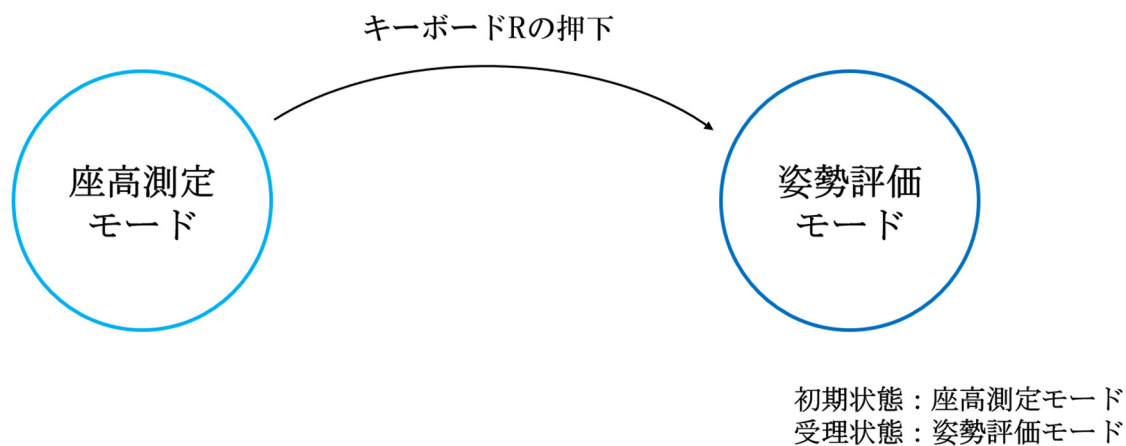


図 44 本システムの状態遷移図

さらに、2つのモードの切り替えは、コード 4 のように実現する。座高測定モードと姿勢評価モードがプログラム上では STATE_INITIALIZING と STATE_MEASURING と定義されている。

```
STATE_INITIALIZING = 0 # 座高設定モードの定義
STATE_MEASURING = 1 # 角度計測モードの定義

state=STATE_INITIALIZING # 座高設定モードでプログラムをスタート
while(True):
    if state== STATE_INITIALIZING:
        # 座高の計算
    elif state== STATE_MEASURING:
        # 角度の計算

    # 座高設定を抜けるための条件
    if keyboard.is_pressed('r'):
        state=STATE_MEASURING
```

コード 4 2つのモードを切り替えるプログラムの疑似コード

6 姿勢評価システムの評価実験 (小澤)

本章では、本姿勢評価システムの評価実験について説明する。

本実験は、姿勢評価システムを用いて姿勢を評価できるかを調査するために行う。我々は、3.2.1 項と 4.2.1 項におけるCC'とEE'の近似式作成実験のデータ作成に参加した人(以下、実験経験者)と参加していない人(以下、実験未経験者)それぞれ2名を用意し、姿勢の変化に伴い、適切な姿勢ガイダンスが表示されるか調べた。

6.1 実験方法 (小澤)

実験経験者2名をそれぞれA、Bとし実験未経験者2名をそれぞれC、Dとする。本実験の評価対象姿勢は、角度が0°時の姿勢を良姿勢、角度が30°~60°時の姿勢を弱悪姿勢、角度が90°時の姿勢を悪姿勢とし、以下を検証する。

- A, B, C, Dの良姿勢において、適切な姿勢ガイダンスが表示されること
- A, B, C, Dの悪姿勢、弱悪姿勢において、適切な姿勢ガイダンスが表示されること

6.2 結果 (小澤)

図45~図56は、被験者A, B, C, Dの各4名の実験中において、姿勢ガイダンスが表示されている様子である。図45~図56の通り、被験者全員で、背筋を伸ばした姿勢(0°)では図40の姿勢ガイダンスが、体を大きく前傾させた姿勢(90°)では図41、図42の姿勢ガイダンスが表示された。また、姿勢の変化が背筋をまっすぐ伸ばした状態と近い弱悪姿勢(30°~60°)について図41、図42の姿勢ガイダンスが表示されたことから、細かな姿勢の変化にも対応可能であることがわかる。

これらの結果より、良姿勢と悪姿勢の変化に伴い適切な姿勢ガイダンスを表示し、本システムを用いて姿勢を評価することに成功した。



図 45 A の良姿勢(0°)における姿勢ガイダンスの表示



図 46 A の弱悪姿勢(30°~60°)における姿勢ガイダンスの表示



図 47 A の悪姿勢(90°)における姿勢ガイダンスの表示



図 48 B の良姿勢(0°)における姿勢ガイダンスの表示

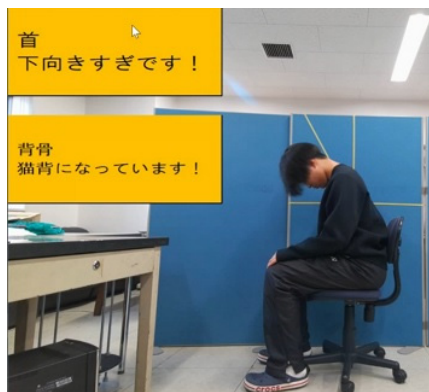


図 49 B の弱悪姿勢(30°~60°)における姿勢ガイダンスの表示



図 50 B の悪姿勢(90°)における姿勢ガイダンスの表示



図 51 Cの良姿勢(0°)における姿勢ガイダンスの表示



図 52 Cの弱悪姿勢(30°~60°)における姿勢ガイダンスの表示



図 53 Cの悪姿勢(90°)における姿勢ガイダンスの表示



図 54 Dの良姿勢(0°)における姿勢ガイダンスの表示



図 55 Dの弱悪姿勢(30°~60°)における姿勢ガイダンスの表示



図 56 Dの悪姿勢(90°)における姿勢ガイダンスの表示

7 結論と課題（小澤）

7.1 結論

本研究では、以下の3つの条件を満たす(姿勢評価システム)の開発を目指した。

1. カメラを用いて人物の特徴を認識できること
2. 定量的な姿勢評価を行えること
3. リアルタイムに姿勢を分析し、悪い姿勢であれば改善策を表示できること

1の条件については、Azure Kinectを導入することにより条件を満たしたと言える。

2については、理学療法を用いた姿勢評価方法を参考にすることで客観性を確保し、胸椎後弯角 θ_{tk} と頸椎前弯角 θ_{cl} を求めたことにより、条件を満たしたと言える。

3については、2つのモードを併用することでリアルタイムでの角度を算出し、その結果に伴い、姿勢ガイダンスを表示できたことから、条件を満たしたと言える。

7.2 課題

本研究の姿勢評価システムにおける評価実験において、測定手法におけるCC'とEE'の近似式作成実験のデータ作成に参加した2名と参加していない2名の合計4名に対しては姿勢評価が可能であることが確認された。しかし、適用範囲を広げるためには、高齢者や円背を有する者、腰痛患者に対する妥当性の検討も必要である。この課題を解決することができれば、姿勢を評価する方法の一つとして今後の活躍が期待できる。

参考文献

- [1] 導入状況の推移 | 基本情報 | テレワーク総合ポータル
<https://telework.mhlw.go.jp/telework/trs/>
- [2] 大学のオンライン授業実施率は97%、導入時期は4-5月が多数 | 教育業界ニュース「ReseEd (リシード)」
<https://reseed.resemom.jp/article/2020/07/17/473.html>
- [3] 【テレワークとなった働き世代1,000人へ緊急アンケート】 新型コロナウイルスによる、働き方・暮らしの変化により「肩こり」「精神的ストレス」などの身体的不調を実感 | オムロンヘルスケア株式会社
<https://www.healthcare.omron.co.jp/corp/news/2020/0428.html>
- [4] オンライン授業に関する調査結果 (2020年度春学期) | 早稲田大学
<https://www.waseda.jp/top/news/70555>
- [5] 高桑 巧, 研谷 智, 柏崎 裕一, “高校生肩こり-アンケート調査より-”, 肩関節, 24巻, 2号, p.195-198, (2000)
https://www.jstage.jst.go.jp/article/katakansetsu1977/24/2/24_195/_article/-char/ja/
- [6] 高井 美咲, 田中 芳幸, “課題遂行時の座位姿勢が心身の急性ストレス反応に及ぼす影響”, 日本健康心理学会大会発表論文集, 33巻, (2020)
https://www.jstage.jst.go.jp/article/jahpp/33/0/33_P02/_article/-char/ja/
- [7] 内藤 孝雄, 黒須 誠治, “デスクワーク時の着座姿勢の評価方法の提案”, 人工知能学会全国大会論文集, 第28回, (2014)
https://www.jstage.jst.go.jp/article/pjsai/JSIAI2014/0/JSIAI2014_3E44in/_article/-char/ja/
- [8] 大村 颯太, 横山 茂樹, “デジタル傾斜計を用いた胸椎後弯角に関する 評価方法の検者内・検者間信頼性”, 理学療法科学, 36巻, 3号, p.325-329, (2021)
https://www.jstage.jst.go.jp/article/rika/36/3/36_325/_article/-char/ja/
- [9] 竹井 仁, “姿勢の評価と治療アプローチ”, 脊髄外科, 27巻, 2号, p.119-124, (2013)
https://www.jstage.jst.go.jp/article/spinalsurg/27/2/27_119/_article/-char/ja/
- [10] 市川 秀男, “X線画像読影に必要な計測法-インナービジョン”, INNERVISION, 35巻, 4号, (2020)
https://www.innervision.co.jp/ressources/pdf/innervision2020/iv202004_068.pdf
- [11] Azure Kinect Body Tracking SDK のダウンロード | Microsoft
<https://www.microsoft.com/en-us/download/details.aspx?id=104221>

[12] Azure Kinect ボディトラッキングの関節 | Microsoft Learn
<https://learn.microsoft.com/ja-jp/azure/Kinect-dk/body-joints>

[13] 有限オートマトンとは | IT を分かりやすく解説
<https://medium-company.com/%e6%9c%89%e9%99%90%e3%82%aa%e3%83%bc%e3%83%88%e3%83%9e%e3%83%88%e3%83%b3/>

[14] オートマトンとは何か？わかりやすく解説します！ | にほんブログ村
<https://www.krrk0.com/automaton/>

謝辞

本研究を進めるにあたり、2年間ご指導していただきました工学院大学 先進工学部 機械理工学科 金丸隆志教授、研究室の皆様と学科の同級生にメンバー一同心より感謝いたします。ご指導いただいた内容を今後に生かせるよう精進いたします。ここに感謝の意を表します。

付録

姿勢評価システムのプログラム

```
"""
This is a python port of the simple-sample project from.
    https://github.com/microsoft/Azure-Kinect-
Samples/blob/bf2f8cf95d969dcc7842c4c450052fe5a943c756/body-tracking-
samples/simple_sample/main.c
"""

import traceback
import sys
import ctypes
import os
import cv2
import keras
import numpy as np
import math
import matplotlib.pyplot as plt
import sympy as sy
from scipy import optimize, exp
import decimal
import time
from time import sleep
import datetime
import keyboard

# Add .. to the import path
sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__),
'..')))

import k4a

def VERIFY(result, error):
    if result != k4a.K4A_RESULT_SUCCEEDED:
```

```

    print(error)
    traceback.print_stack()
    sys.exit(1)

def print_body_information(body):
    print("Body ID: {}".format(body.id))
    for i in range(k4a.K4ABT_JOINT_COUNT):
        position = body.skeleton.joints[i].position
        orientation = body.skeleton.joints[i].orientation
        confidence_level = body.skeleton.joints[i].confidence_level
        print("Joint[{}]: Position[mm] ( {}, {}, {} ); Orientation ( {}, {}, {},
{}); Confidence Level ({} )".format(
            i, position.v[0], position.v[1], position.v[2], orientation.v[0],
orientation.v[1], orientation.v[2], orientation.v[3], confidence_level
        ))

def print_body_index_map_middle_line(body_index_map):
    print("print_body_index_map_middle_line not implemented")
    """
    uint8_t* body_index_map_buffer = k4a_image_get_buffer(body_index_map);

    // Given body_index_map pixel type should be uint8, the stride_byte should
be the same as width
    // TODO: Since there is no API to query the byte-per-pixel information, we
have to compare the width and stride to
    // know the information. We should replace this assert with proper byte-per-
pixel query once the API is provided by
    // K4A SDK.
    assert(k4a_image_get_stride_bytes(body_index_map) ==
k4a_image_get_width_pixels(body_index_map));

    int middle_line_num = k4a_image_get_height_pixels(body_index_map) / 2;
    body_index_map_buffer = body_index_map_buffer + middle_line_num *
k4a_image_get_width_pixels(body_index_map);

    printf("BodyIndexMap at Line %d:\n", middle_line_num);

```

```

for (int i = 0; i < k4a_image_get_width_pixels(body_index_map); i++)
{
    printf("%u, ", *body_index_map_buffer);
    body_index_map_buffer++;
}
printf("\n");
"""

if __name__ == "__main__":

    if len(sys.argv)==1:
        print('Usage: python recognize_skeleton.py savedfile.h5')
        sys.exit()
    savedfile = sys.argv[1]
    model = keras.models.load_model(savedfile)

    num_joints = 32
    num_features = num_joints * 3 # 32 joints * 3 dimensions

    data = np.empty((num_joints, 3), float)

    device_config = k4a.K4A_DEVICE_CONFIG_INIT_DISABLE_ALL
    device_config.depth_mode = k4a.K4A_DEPTH_MODE_NFOV_UNBINNED

    device = k4a.k4a_device_t()
    VERIFY(k4a.k4a_device_open(0, ctypes.byref(device)), "Open K4A Device
failed!")
    VERIFY(k4a.k4a_device_start_cameras(device, ctypes.byref(device_config)),
"Start K4A cameras failed!")

    sensor_calibration = k4a.k4a_calibration_t()
    VERIFY(k4a.k4a_device_get_calibration(device, device_config.depth_mode,
k4a.K4A_COLOR_RESOLUTION_OFF, ctypes.byref(sensor_calibration)), "Get depth
camera calibration failed!")

```

```

tracker = k4a.k4abt_tracker_t()
tracker_config = k4a.K4ABT_TRACKER_CONFIG_DEFAULT
try:
    ctypes.cdll.LoadLibrary("C:/Program Files/Azure Kinect Body Tracking
SDK/tools/directml.dll")
except Exception as e:
    tracker_config.processing_mode =
k4a.K4ABT_TRACKER_PROCESSING_MODE_GPU_CUDA
    #tracker_config.processing_mode = k4a.K4ABT_TRACKER_PROCESSING_MODE_CPU
    VERIFY(k4a.k4abt_tracker_create(ctypes.byref(sensor_calibration),
tracker_config, ctypes.byref(tracker)), "Body tracker initialization failed!")

cap = cv2.VideoCapture(0)

while True:
    sensor_capture = k4a.k4a_capture_t()
    get_capture_result = k4a.k4a_device_get_capture(device,
ctypes.byref(sensor_capture), k4a.K4A_WAIT_INFINITE)
    ret, frame = cap.read()

    if get_capture_result == k4a.K4A_WAIT_RESULT_SUCCEEDED:
        #frame_count += 1

        #print("Start processing frame {}".format(frame_count))

        queue_capture_result = k4a.k4abt_tracker_enqueue_capture(tracker,
sensor_capture, k4a.K4A_WAIT_INFINITE)

        k4a.k4a_capture_release(sensor_capture)

    if queue_capture_result == k4a.K4A_WAIT_RESULT_TIMEOUT:
        # It should never hit timeout when K4A_WAIT_INFINITE is set.
        print("Error! Add capture to tracker process queue timeout!")

```

```

        break
    elif queue_capture_result == k4a.K4A_WAIT_RESULT_FAILED:
        print("Error! Add capture to tracker process queue failed!")
        break

    body_frame = k4a.k4abt_frame_t()
    pop_frame_result = k4a.k4abt_tracker_pop_result(tracker,
ctypes.byref(body_frame), k4a.K4A_WAIT_INFINITE)
    if pop_frame_result == k4a.K4A_WAIT_RESULT_SUCCEEDED:
        num_bodies = k4a.k4abt_frame_get_num_bodies(body_frame)
        #print("{} bodies are detected!".format(num_bodies))
        for i in range(num_bodies):
            body = k4a.k4abt_body_t()
            VERIFY(k4a.k4abt_frame_get_body_skeleton(body_frame, i,
ctypes.byref(body.skeleton)), "Get body from body frame failed!")
            body.id = k4a.k4abt_frame_get_body_id(body_frame, i)

            #print_body_information(body)

            img1 = cv2.imread('b1.png')
            img3 = cv2.imread('c1.png')
            imggood = cv2.imread('good.png')
            height,width = img1.shape[:2]

STATE_INITIALIZING = 0 # 座高設定モードの定義
STATE_MEASURING = 1 # 角度計測モードの定義

state=STATE_INITIALIZING # 座高設定モードでプログラムをスタート

# 座高設定を抜けるための条件
if keyboard.is_pressed('r'):
    state=STATE_MEASURING

if i==0:
    if state==STATE_INITIALIZING:

```

```

head_x = body.skeleton.joints[26].position.xyz.x
head_y = body.skeleton.joints[26].position.xyz.y
head_z = body.skeleton.joints[26].position.xyz.z

neck_x = body.skeleton.joints[3].position.xyz.x
neck_y = body.skeleton.joints[3].position.xyz.y
neck_z = body.skeleton.joints[3].position.xyz.z

pelvis_x = body.skeleton.joints[0].position.xyz.x
pelvis_y = body.skeleton.joints[0].position.xyz.y
pelvis_z = body.skeleton.joints[0].position.xyz.z

naval_x = body.skeleton.joints[1].position.xyz.x
naval_y = body.skeleton.joints[1].position.xyz.y
naval_z = body.skeleton.joints[1].position.xyz.z

chest_x = body.skeleton.joints[2].position.xyz.x
chest_y = body.skeleton.joints[2].position.xyz.y
chest_z = body.skeleton.joints[2].position.xyz.z

#head から neck
HN = math.sqrt((head_x - neck_x)**2 + (head_y -
neck_y)**2 + (head_z - neck_z)**2)
#neck から chest
NC = math.sqrt((neck_x - chest_x)**2 + (neck_y -
chest_y)**2 + (neck_z - chest_z)**2)
#chest から naval
CN = math.sqrt((chest_x - naval_x)**2 + (chest_y -
naval_y)**2 + (chest_z - naval_z)**2)
#naval から pelvis
NP = math.sqrt((naval_x - pelvis_x)**2 + (naval_y -
pelvis_y)**2 + (naval_z - pelvis_z)**2)
#座高 z
z = (HN + NC + CN + NP)
CC = 0.0726*z - 1.5365

```



```

EE = 0.3272*z - 3.0457
print("座高の長さ="+str(round(z, 2)))

elif state==STATE_MEASURING:
    head_x = body.skeleton.joints[26].position.xyz.x
    head_y = body.skeleton.joints[26].position.xyz.y
    head_z = body.skeleton.joints[26].position.xyz.z

    neck_x = body.skeleton.joints[3].position.xyz.x
    neck_y = body.skeleton.joints[3].position.xyz.y
    neck_z = body.skeleton.joints[3].position.xyz.z

    naval_x = body.skeleton.joints[1].position.xyz.x
    naval_y = body.skeleton.joints[1].position.xyz.y
    naval_z = body.skeleton.joints[1].position.xyz.z

    chest_x = body.skeleton.joints[2].position.xyz.x
    chest_y = body.skeleton.joints[2].position.xyz.y
    chest_z = body.skeleton.joints[2].position.xyz.z

    # 上半身
    a_x = (3 * head_x + 2 * chest_x) / 5
    a_y = (3 * head_y + 2 * chest_y) / 5
    a_z = (3 * head_z + 2 * chest_z) / 5

    b_x = (2 * naval_x + 1 * chest_x) / 3
    b_y = (2 * naval_y + 1 * chest_y) / 3
    b_z = (2 * naval_z + 1 * chest_z) / 3

    c_x = (37 * b_x + 46 * a_x) / 83
    c_y = (37 * b_y + 46 * a_y) / 83
    c_z = (37 * b_z + 46 * a_z) / 83

    C_x = (CC/(math.sqrt((a_z - b_z)**2 + (b_x -
a_x)**2)))*(a_z - b_z) + c_x
    C_y = c_y

```

$$C_z = (CC / (\sqrt{(a_z - b_z)^2 + (b_x - a_x)^2})) * (b_x - a_x) + c_z$$

$$b_x = b_x - C_x$$

$$b_y = b_y - C_y$$

$$b_z = b_z - C_z$$

$$a_x = a_x - C_x$$

$$a_y = a_y - C_y$$

$$a_z = a_z - C_z$$

$$\cos = \frac{(a_x b_x + a_y b_y + a_z b_z)}{(\sqrt{a_x^2 + a_y^2 + a_z^2}) * (\sqrt{b_x^2 + b_y^2 + b_z^2})}$$

$$\cos 180 = (-1)$$

$$ac = \sqrt{(a_x - c_x)^2 + (a_y - c_y)^2 + (a_z - c_z)^2}$$

$$bc = \sqrt{(b_x - c_x)^2 + (b_y - c_y)^2 + (b_z - c_z)^2}$$

$$\alpha = \arctan(ac/CC)$$

$$\beta = \arctan(bc/CC)$$

$$\theta_{rad} = \alpha + \beta$$

$$\theta_1 = \text{degrees}(\theta_{rad})$$

$$\theta_{2rad} = \arccos(\cos 180)$$

$$\theta_2 = \text{degrees}(\theta_{2rad})$$

$$\theta_{TK} = \theta_2 - \theta_1$$

#頸椎

$$d_x = \text{head}_x$$

$$d_y = \text{head}_y$$

$$d_z = \text{head}_z$$

```

a_x = (3 * head_x + 2 * chest_x) / 5
a_y = (3 * head_y + 2 * chest_y) / 5
a_z = (3 * head_z + 2 * chest_z) / 5

e_x = (7 * a_x + 29 * d_x) / 36
e_y = (7 * a_y + 29 * d_y) / 36
e_z = (7 * a_z + 29 * d_z) / 36

de = math.sqrt((d_x - e_x)**2 + (d_y - e_y)**2 + (d_z -
e_z)**2)

ae = math.sqrt((a_x - e_x)**2 + (a_y - e_y)**2 + (a_z -
e_z)**2)

alpha = math.atan(de/EE)
beta = math.atan(ae/EE)
theta7rad = alpha + beta
theta_CL = math.degrees(theta7rad)

print('胸椎=' + str(round(theta_TK, 2)))
print("頸椎=" + str(round(theta_CL, 2)))

#頸椎
if 31.4 < theta_CL < 33.4:
    frame[0:height, 0:width] = imggood
elif 31.4 > theta_CL:
    frame[0:height, 0:width] = img1

#胸椎
if 35 < theta_TK < 36.2:
    frame[200:height + 200, 0:width] = imggood
elif 36.2 < theta_TK:
    frame[200:height + 200, 0:width] = img3

```

```

        body_index_map = k4a.k4abt_frame_get_body_index_map(body_frame)
        if body_index_map:
            #print_body_index_map_middle_line(body_index_map)
            k4a.k4a_image_release(body_index_map)
        else:
            print("Error: Fail to generate bodyindex map!")

        k4a.k4abt_frame_release(body_frame)
    elif pop_frame_result == k4a.K4A_WAIT_RESULT_TIMEOUT:
        # It should never hit timeout when K4A_WAIT_INFINITE is set.
        print("Error! Pop body frame result timeout!")
        break
    else:
        print("Pop body frame result failed!")
        break

    elif get_capture_result == k4a.K4A_WAIT_RESULT_TIMEOUT:
        # It should never hit timeout when K4A_WAIT_INFINITE is set.
        print("Error! Get depth frame time out!")
        break
    else:
        print("Get          depth          capture          returned          error:
{}").format(get_capture_result))

    cv2.imshow('fram 点 E' , frame)

    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break

print("Finished body tracking processing!")

k4a.k4abt_tracker_shutdown(tracker)
k4a.k4abt_tracker_destroy(tracker)

```

```
k4a.k4a_device_stop_cameras(device)
```

```
k4a.k4a_device_close(device)
```