

# Kinect を利用したゼスチャーによる プログラム入力の可能性

大西 修平<sup>1,a)</sup> 野部 緑<sup>1,2,b)</sup> 中野 由章<sup>1,c)</sup> 兼宗 進<sup>1,d)</sup>

**概要:** 教育用プログラミング言語ドリトルの処理系を、マイクロソフト社の入力デバイスである Kinect の計測値を読み取れるように拡張した。現在はフリーソフトの KineX を介して身体の関節ごとの位置座標をネットワーク経由で取得しているが、ドリトルから Kinect への直接的な接続についても調査を進めている。これらの実装とともに、身体によるポーズを利用したプログラム入力の可能性について報告する。

**キーワード:** ドリトル, プログラミング言語, Kinect, KineX

## Program Input using Kinect in Dolittle Program Environment

OONISHI SHUHEI<sup>1,a)</sup> NOBE MIDORI<sup>1,2,b)</sup> NAKANO YOSHIAKI<sup>1,c)</sup> KANEMUNE SUSUMU<sup>1,d)</sup>

**Abstract:** We have developed a new feature of Dolittle programming environment to communicate with the Microsoft Kinect device. Now we use the Kinect support software KineX that send coordinate data to network periodically. We report the implementation and possibility of program input by physical gesture.

**Keywords:** Dolittle, programming language, kinect

### 1. はじめに

情報技術の発達により、入力デバイスが高性能化が進み、従来は不可能であった形の入力や計測が可能になりつつある。その一方で、情報機器の利用は生活の中に広がっており、情報端末を使える人と使えない人の情報格差も指摘されている。

我々は身体の位置情報を測定するデバイスである Kinect に着目した。Kinect を用いると、キーボードやマウスを用いることなく、身体的な動きをコンピュータに伝えること

ができる。この特徴を利用することで、キーボードなどを扱えない幼児や高齢者、身体障害者などに入力の代替手段を提供できる可能性がある。

今回は、Kinect を利用して、プログラム入力に関する予備的な実験を行った。まず、プログラミング言語ドリトルの処理系から、マイクロソフト社の入力デバイスである Kinect の値を読み取れるように拡張した。現在はフリーソフトの KineX を介して身体の関節ごとの位置座標をネットワーク経由で取得している。取得した値を利用して、ロボットカーの制御を行うためのプログラムを身体的なポーズによって入力するシステムを開発した。システムは実際に大学のオープンキャンパスで使用し、高校生が利用できることを確認した。本論文ではその概要を報告し、ポーズやゼスチャーによる入力の利用可能性について議論を行う。

### 2. 使用した技術

#### 2.1 Kinect

Kinect[4] はマイクロソフト社が開発したゲーム機(XBox)

<sup>1</sup> 大阪電気通信大学  
Osaka Electro-Communication University, Neyagawa, Osaka  
572-8530, Japan

<sup>2</sup> 大阪府立寝屋川高校  
Neyagawa Upper Secondary School, Neyagawa, Osaka 572-  
0832, Japan

a) ml12a004@oecu.jp

b) midori@mbox.kyoto-inet.or.jp

c) info@nakano.ac

d) kanemune@acm.org

用の入力デバイスである。赤外線をマトリクス状に照射して撮影することで、物体の深度情報を取得したり、背景から人体を識別して取得することが可能である。可視光の撮影も可能であり、赤外線で検出した画像と合成して扱うことが可能である。

## 2.2 KineX

KineX (キネックス) は、Kinect の骨格データをネットワーク経由で放流するソケットサーバである [1]。Kinect で認識した関節情報を、1 秒間に 30 回程度の頻度で、ネットワーク上に繰り返し送信する。

データ形式は HTML や XML 風のタブで記述されたテキストデータあり、人体の「頭」「右手首」などの関節について、空間での 3 次元座標 (x, y, z) と、画面上の 2 次元座標 (sx, sy) が含まれている。たとえば、頭部 (Head) の座標は「<Head> x y z sx sy」という形式で表現される。

通常、Kinect から骨格を認識するためにはキャリブレーション (骨格判定のための初期ポーズ) が必要であるが、それを不要にしていることも利点である。

図 1 に KineX の動作画面を示す。画面の左側に画像が表示され、認識された人体の関節が表示される。画面の右側では送信する関節を選択する。標準ではすべての関節情報が送信される。画面の下には、送信する IP アドレスとポート番号 (8888) が表示されており、Start ボタンで送信が開始される。図 2 に KineX から送出される座標データの例を示す。

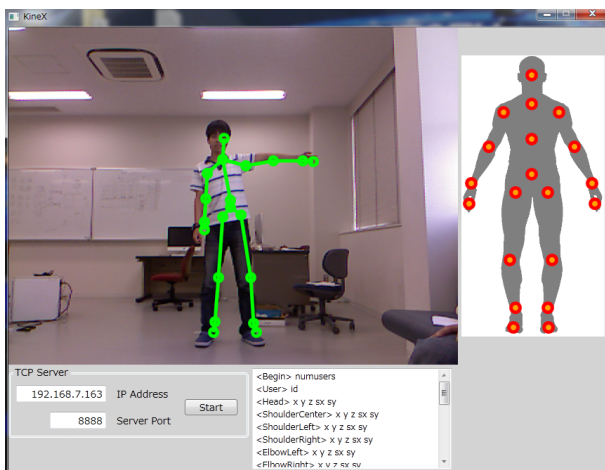


図 1 KineX の動作画面

## 2.3 ドリトルの KineX オブジェクト

ドリトルは教育用のオブジェクト指向言語である。KineX から送出される座標データを取得するために、V2.32 から新たに KineX オブジェクトにアクセスする機能を用意した。図 3 に KineX オブジェクトを使用したプログラム例を示す。

```
<Begin> 1
<User> 254
<Head> -0.19 0.53 2.91 282 165
<ShoulderCenter> -0.20 0.35 2.94 280 197
<ShoulderLeft> -0.36 0.26 2.96 249 215
<ShoulderRight> -0.04 0.24 2.92 307 217
<ElbowLeft> -0.42 0.01 2.93 240 257
<ElbowRight> 0.01 -0.02 2.88 316 265
<WristLeft> -0.44 -0.21 2.83 232 299
<WristRight> 0.06 -0.21 2.79 327 300
<HandLeft> -0.45 -0.30 2.80 230 316
<HandRight> 0.08 -0.30 2.77 331 318
<Spine> -0.20 0.01 2.90 279 259
<HipCenter> -0.20 -0.05 2.84 278 268
<HipLeft> -0.27 -0.11 2.83 264 281
<HipRight> -0.13 -0.12 2.81 292 281
<KneeLeft> -0.29 -0.55 2.79 260 362
<KneeRight> -0.09 -0.58 2.80 297 367
<AnkleLeft> -0.29 -0.88 2.78 258 427
<AnkleRight> -0.09 -0.91 2.79 297 432
<FootLeft> -0.29 -0.96 2.74 258 445
<FootRight> -0.09 -0.98 2.76 297 447
<End>
```

図 2 KineX から送出される座標データの例

KineX オブジェクトは「接続」メソッドにより起動し、パラメータで指定された IP アドレスの 8888 ポートの監視を開始する。

**kinex!"192.126.1.24" 接続。**

同じマシン内で KineX に接続する場合は、IP アドレスは次のように書くことも可能である。

**kinex! (システム:ipaddress) 接続。**

KineX に接続した KineX オブジェクトは、データが到着するたびにそれらを受け取り、オブジェクト内に保持する。そして、ユーザーのプログラムから「読む」メソッドを送ることで最新のデータを取り出すことが可能である。

**ret = kinex !読む。**

このプログラムを実行すると、戻り値として、関節ごとの x, y, z, sx, sy がプロパティとして格納されたオブジェクトが返される。たとえば、頭部 (Head) の x 座標は、次のように取り出すことが可能である。

**x = ret : 頭x。**

**kinex!(システム:ipaddress) 接続。**

表示欄=ラベル!作る。

「

関節 = kinex !読む。

x = 関節 : 右手首x。

表示欄! (x) 書く。

」! 100 回 繰り返す。

図 3 KineX と通信するドリトルのプログラム例

### 3. ポーズを判定するメソッドの追加

KineX オブジェクトを利用することで、ドリトルのプログラムから Kinect の座標を扱うことができるようになったが、具体的なポーズを座標で判定することは利用者にとって敷居が高い。そこで、中学校や高校の授業で使えるようなポーズを判定するメソッドを用意した。

表 1 に、ドリトルから利用できるポーズの一覧と、それぞれの判定条件の例を示す。実際には、関節同士の大小関係だけでなく、「左ひざが右ひざより 5cm 以上上にある」のような数値定数を加味した判断を行なっている。

表 1 ポーズを判定するドリトルのメソッド例

命令	判定
右拳手?	(右手首y > 右ひじy) (右手首y < 頭y)
左拳手?	(左手首y > 左ひじy) (左手首y < 頭y)
両手拳手?	(右拳手?) && (左拳手?)
右足上?	右ひざy > 左ひざy
左足上?	左ひざy > 右ひざy
右手上?	右手首y > 頭y
左手上?	左手首y > 頭y
両手上?	(右手上?) && (左手上?)
右手左?	右手首x < 腹x
左手右?	左手首x > 腹x
両手クロス?	(右手左?) && (左手右?)
右手横?	右手首x > 右肩x
左手横?	左手首x < 左肩x
両手横?	(右手横?) && (左手横?)
右手前?	右手首z < 右肩z
左手前?	左手首z < 左肩z
両手前?	(右手前?) && (左手前?)

図 4 に、ポーズ判定命令を使用したドリトルプログラムの例を示す。XY 座標を使わなくても、拳手しているかどうかといった自然な命令でポーズを判定できる。

「関節! 右拳手?!」!なら「表示欄! (x) 書く」実行。

```

kinex!(システム:ipaddress) 接続。
表示欄=ラベル! 作る。
「
    関節 = kinex ! 読む。
    「関節! 右拳手?!」!なら「表示欄! (x) 書く」実行。
」! 100 回 繰り返す。

```

図 4 ポーズ判定命令を使用したプログラム例

これらのメソッドを作成する過程で、Kinect から人間の自然なポーズを判定するために、いくつかの難しい部分が明らかになった。今後も引き続き、安定して使用できるメソッドの調査が必要と考えている。

- 人間は手の指の動きが重要だが、関節は手の甲や手首といった精度であり、個々の指の識別や指先の位置の

判定はできない。

- 足は上半身と違い、位置を比較できる関節が近くはない。足を上げる動作の判定は「左右の膝の Y 座標が 5cm 以上違う」といった条件で可能だが、この差分は身長などで異なる。(頭とつま先から身長を推定して補正すべきか?)
- 足は、長いスカートを穿いている場合などは正しく認識されない場合があった。
- 関節のいくつかは、身体での位置が曖昧な場合がある。たとえば、頭 (Head) は画面では「ひたい」「鼻」「口」の辺りに見えるし、背骨 (Spine) は「みぞおち」「へそ」の辺りに見えるが、ポーズを作る人にわかりづらい。

### 4. ポーズによるプログラム入力

Kinect による入力を利用する例として、身体的なポーズでプログラムを入力するシステムを試作した。題材はロボットカーを平面上で前後に動かす動作と、左右に回転させる動作とした。簡単のために、移動距離や回転角度などのパラメータは固定とし、命令のみを入力する形とした。

命令は「前進」「左回転」「後退」「右回転」「終了」の 5 種類である。命令のポーズを図 5 に示す。図はポーズを取る操作者が見やすいように鏡像の画面になっており、すべて右手を移動させるポーズの画面である。

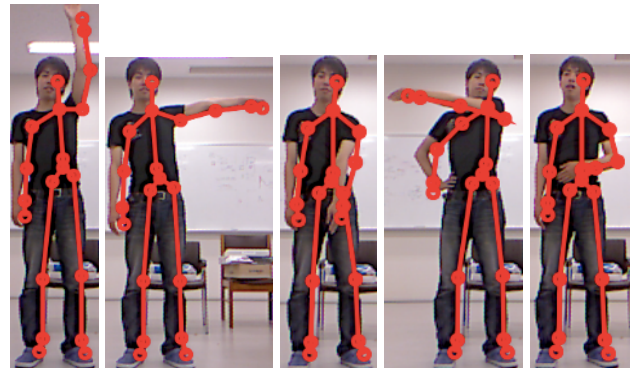


図 5 命令のポーズ

今回作成したシステムの処理の流れを図 6 に示す。Kinect で検出した身体の X,Y 座標を KineX で読み取り、ドリトルに転送する。<sup>\*1</sup> ドリトルでは、関節の座標からポーズを判定して命令を解釈し、ロボットカーに転送するプログラムとして登録する。

ポーズの判定は、当初は撮影した画面の中で、右手首の XY 座標で判定するように実装したが、立つ位置や身長の違いなどの影響により必ずしも毎回右手首の位置が同じ座標にならず、正しく動作させることができなかつた。

そこで、右手首の座標を他の関節の座標と比較すること

<sup>\*1</sup> 試作の際は図のように KineX とドリトルを接続する中継サーバーを Java で記述して使用した。

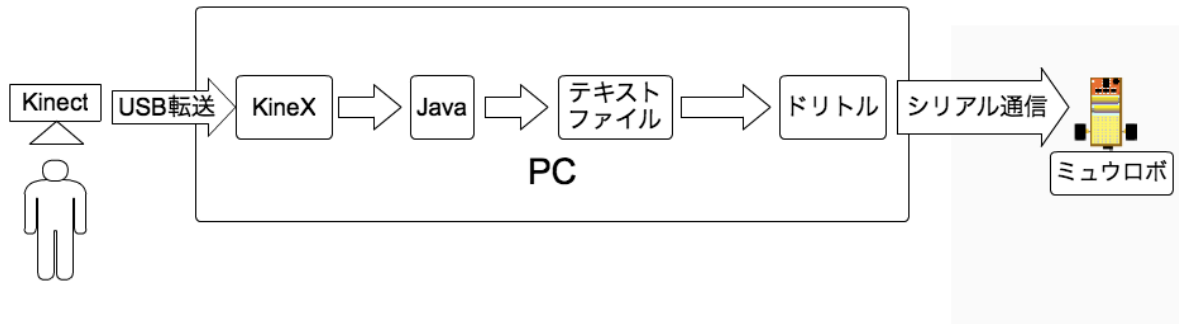


図 6 処理の流れ

で改善を試みた。命令を判断する位置関係を図 7 に示す。この改良により、前進は「右手首の Y 座標が右肩の Y 座標より大きい」、後退は「右手首の Y 座標が右腰の Y 座標より小さい」、右回転は「右手首の X 座標が右肩の X 座標より大きい」、左回転は「右手首の X 座標が左肩の X 座標より小さい」という判断が可能になり、立つ位置や身長などの違いによらず安定した判定が可能になった。

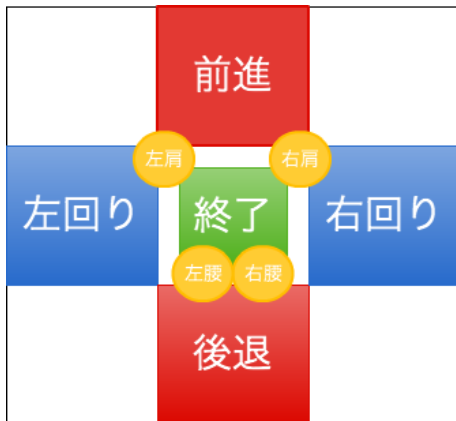


図 7 右手首の位置と命令との対応

試作したシステムを、大学のオープンキャンパスに参加した高校生に使ってもらった。図 8 は、ポーズによりプログラムを入力している様子である。画面には自分の画像と、認識された命令を表示している。ポーズは 5 秒ごとに撮影するようにした。

入力されたプログラムはドリトルのプログラムとして取り込み、ロボットカーに転送する。図 9 に、認識された命令と、ロボットカーに転送する様子を示す。

オープンキャンパスでは、2 日間で約 30 人に使ってもらうことができた。反応は概ね好評であり、参加者からは「身体を使って入力できることに驚いた」「画像を認識する技術が高度だと思った」「プログラムが自分でも簡単に書けた」「体の動きがロボットカーの動きになることが楽しかった」などの感想があった。

システム的には、いくつかの課題が明らかになった。

- ポーズで命令を入力していくが、少し長い命令を入力すると今まで入れた命令がわからなくなり、どのよう

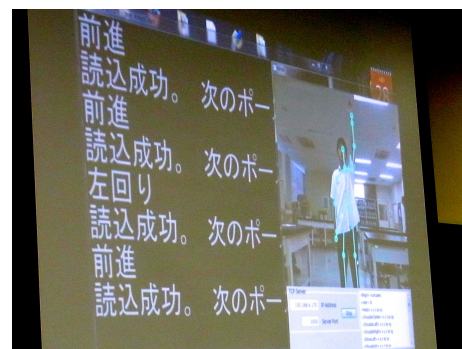
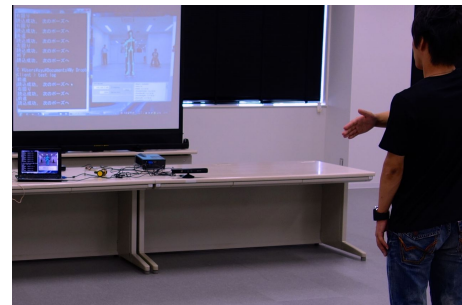


図 8 ポーズによるプログラム入力の様子

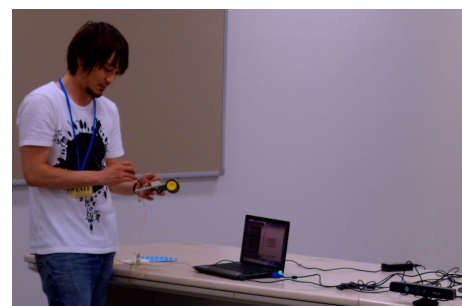
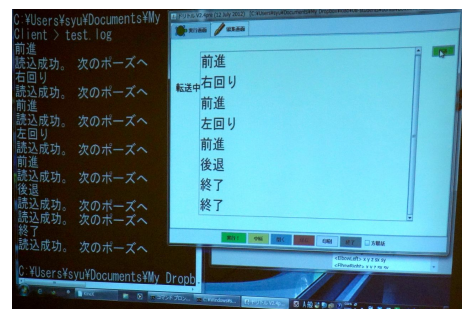


図 9 入力されたプログラムとロボットカーへの転送

に動くかが予想できなくなってしまっていた。あらかじめ動きを紙などで考えさせる指導が必要だった。

- 入力した命令を訂正する手段がなかった。命令を誤って入力すると、そのままロボットカーに転送される。命令の誤入力はその場で気づくので、最後の命令を取り消すポーズを追加することで改善できると考えられる。
- システムを開始するとポーズ入力が開始されてしまう。プログラム入力の終了命令だけでなく開始命令も用意することで操作性が改善されることが考えられる。

システムを試作し、高校生を対象に実験を行った。今後は11月に、小学生向けのイベントで使用する予定である。上記の課題をふまえて改良を進めたいと考えている。

## 5. Kinect のプログラミング利用の可能性

今回利用した Kinect について、今後プログラミング教育や情報教育においてどのような利用の可能性があるかを検討する。

### 5.1 CS アンブラグドとコンピュータのかけ橋

CS アンブラグドはカードなどの教具を使いながらコンピュータ科学を体験的に理解する学習法である [5]。その中には、身体的な活動が多く含まれている。Kinect を使うことで、CS アンブラグドの学習活動をサポートできる可能性がある。

二進数の学習では、2人が横に並び、4本の手を挙げる動作で4ビットを表現できる。0から15までの16通りの数が画面に表示されることで、2進数が確かにコンピュータに解釈されていることを実感を持って理解できる可能性がある。

画像の符号化では、符号化したモノクロ二値画像を手旗信号のように送信すると、それを画像から読み取り、復号を行った絵を画面に表示する。画像を数値で伝達することで、確かに画像が数値としてコンピュータに扱われていることを実感を持って理解できる可能性がある。

### 5.2 ジェスチャーでのパソコン操作

現在はキーボードやマウスでPCを操作しているが、キーボードを打てない人や、手指を思うように動かせない障害者の方がPCを利用することは困難であり、情報格差につながっている。

Kinect などの入力デバイスを使用して、わずかな体の動きでコンピュータを操作する仕組みは、身体と入力デバイスを接触させる必要もなく、負荷が低い可能性がある。

関連研究として、舌の動きを検知する研究 [6]、人の動きを検知したプレゼンテーションシステム [7]、手の形状の検出 [8]、などがある。

### 5.3 自作プログラムからのセンサー利用

Kinect はゲーム機を操作するために開発された。プログラミングの授業においても、作成したゲーム等のプログラムを操作するための入力デバイスとして Kinect を利用することは可能である。

マウス等のポインティングデバイスの代替としてキャラクタなどを操作したり、ラケットなどを操作する用途が考えられる。

### 5.4 自作制御プログラムからのセンサー利用

ロボットを使用した制御プログラムを学習する際は何らかのセンサーを用いることで位置や傾き、障害物などを検知する。Kinect も外界を検出するデバイスとして使用することが可能である。

関連研究として、Kinect を搭載したロボットカー [9]、などがある。

### 5.5 身体性の測定

Kinect はソフトウェアを制御するコントローラーとしての利用だけでなく、身体の動作や特徴を検出する用途に利用することが可能である。

技能の判定としては、腕の動きや体の動きなどを時系列的に記録することで、ある作業に対する技能を判定できる可能性がある。包丁でのみじん切り、ハンダ付け、プラモデルの組み立て、家具の組立など、多くの用途が考えられる。

動きの判定としては、ヨガやラジオ体操などのポーズを採点するシステムが考えられる。重要な動きやポーズを、たとえば「肩と肘と手首が一直線にあるかどうか」で腕がまっすぐに伸びているかを検出することなどが考えられる。

高齢者の運動支援としては、簡単な体操などのゲームを観察することで認知面や身体面の能力の低下を検出したリ、リハビリに利用する用途などが考えられる。

関連研究として、Kinect を利用した人間の演奏との同期 [10]、Kinect を利用した動作検出 [11]、座位姿勢の検出 [12]、手指形状の検出 [13]、などがある。

## 6. まとめ

教育用プログラミング言語「ドリトル」の Kinect 対応機能と、それを利用したプログラム入力システムを紹介した。Kinect を利用することで、本来のゲーム機の入力機能だけでなく、プログラミング教育や情報教育での利用が可能である。

今回はキーボード等の入力のできない子供や障害者の方を想定し、ポーズやジェスチャーによるプログラム入力システムを設計した。試作したシステムを高校生に使ってもらったところ、実用性を確認できるとともに、いくつかの改良すべき点が明らかになった。今後はシステムの改良を

進めつつ、子ども向けのイベントなどで活用できるように改善を進めていきたい。

本研究は、科学研究費補助金 (基盤研究 (C)) 22500828 の補助を受けています。

## 参考文献

- [1] 工学ナビ. KineX. <http://kougaku-navi.net/kinex/>
- [2] 山石忠弘, 林敏浩, 垂水浩幸. 踊りによるプログラム処理の表現方法. 情報処理学会研究報告, Vol.2010-CE-107, No.12, 2010.
- [3] 教育用プログラミング言語「ドリトル」  
<http://dolittle.eplang.jp>
- [4] 西林孝, 小野憲史. キネクトハッカーズマニュアル.
- [5] 兼宗 進ほか: コンピュータを使わない情報教育アンプラグドコンピュータサイエンス, イーテキスト研究所, 2007.
- [6] 木村堯, 野嶋琢也. SITA:深度情報を利用した舌運動検出手法. 情報処理学会, 第 15 回インタラクション 2012, 2012.
- [7] 前田晴己, 栗原一貴, 宮下芳明. 発表者がスクリーン前に立つプレゼンテーションスタイルの提案. 情報処理学会, エンタテインメントコンピューティング 2011 予稿集, pp.431-434, 2011.
- [8] 岸根和博. IR センサ出力解析による手の形状と動きの検出およびポインティング操作への応用.2011.  
<http://www.ike.tottori-u.ac.jp/thesis/H23.M2/CA.M10T2015A>
- [9] ZMP. RoboCar 1/10 Lite & Kinect.  
<http://www.zmp.co.jp/e-nuvo/jp/robocar-110lite.html>
- [10] 糸原達彦ほか. マルチロボットによる Kinect を用いた同期合奏. 人工知能学会, AI チャレンジ研究会, 2011.
- [11] 小林健太, 紅林秀治. KINECT センサーを用いた人体動作分析システムの開発. 情報処理学会, 情報教育シンポジウム 2012, 2012.
- [12] 藤巻吾郎, 石郷祐介, 成瀬哲哉, 宮川成門, 窪田直樹. 座位姿勢における人体形状測定システムの開発. 岐阜県生活技術研究所研究報告, No.14, 2011.
- [13] 松島慎亮, 桑原教彰, 森本一成. kinect を用いた手指形状の 3 次元計測の一手法. 平成 23 年度日本人間工学会関西支部大会講演論文集, pp.107-108, 2011.