**研究論文**

# Kinect センサーを用いた音声対話システム前における人物の動作識別

早川 元貴*，實廣 貴敏*

(2019 年 9 月 30 日　受理)

# Recognition of person's movement at the front of speech dialog system by a Kinect sensor

Genki Hayakawa*, Takatoshi Jitsuhiro*

(Received September 30, 2019)

We introduce a method to recognize user's movement in front of the spoken dialog system for its autonomous actions. Usual spoken dialog systems just wait for users' utterance. However, if they can talk to users more friendly, users may speak to the systems more happily. In this paper, we specifically consider to detect the four kind of movement patterns that include user's approaching, receding, passing, and stopping. A Kinect sensor developed by Microsoft corp. can be used to obtain user's position and distance from Kinect sensor's position. We use the velocity to recognize whether a user is moving, or stopping, and the direction of velocity to find whether a user is approaching, or receding. Experimental results show that the F-measure, about 80 to 90% was obtained for detecting of approaching, receding, and passing.

**キーワード**： 音声対話システム，Kinect sensor，動作認識，歩行方向認識

**Keywords**： spoken dialog system, Kinect sensor, motion recognition, walking direction recognition

## 1. はじめに

近年，音声対話システムは，スマートフォンやパソコン上で使えるものや，スピーカの形で利用できるものが世間にも広まりつつある．研究としては，古くから構築され，文献[1]などのように，長期間，実際に使われているものもある．著者らが所属する研究室では，大学の施設案内やキャラクターとお話ができる設置型の音声対話システムがある[2].

しかし，これらはどれもユーザーから話しかけるまで何も動作しない．ポスターなどをおいて，アピールしながら，展示しておいても，興味があって近づいてきたとしても，利用する人は少なく，通り過ぎていく人が多い．

対話を行う機会を増やすためには，時にはシステム側からの問いかけがあるとよいが，むやみに話しかけても意味がない．ユーザーの状況を理解した上で話しかけると効果的と考えられる．据え置いた対話システムに興味があれば，近づこうとするだろうし，そうでなければ，通り過ぎる．これらの動作を検出できれば，話しかけるタイミングを図る参考になると考えられる．

そこで，近づいてくる人を検出することができれば，システム側から挨拶や呼び込みを行うことでシステムの利用率が上がると考えられる．本研究では，ユーザーの位置情報を取得することのできる Microsoft 社製の Kinect センサーを用いて，ユーザーの動きを検出し，近づいているのか，離れていっているのかなどの動作を検出することを目標とする．この Kinect センサーを利用した関連研究として，Kinect センサーの骨格検出によりユーザーの頭部を追跡し位置情報の取得を行い，この位置情報にマイクの指向性を向けることでユーザーの音声を強調する手法が

*愛知工科大学工学部情報メディア学科 愛知県蒲郡市西迫町馬乗 50-2
*Department of Media Informatics, Faculty of Engineering, Aichi University of Technology, 50-2, Manori, Nishihasama-cho, Gamagori-shi, Aichi, 443-0047 Japan

ある[3]. その結果, 雑音による影響が少なく, 安定した音源の位置推定が行える. また, 講義中の受講者の状態を推定するときに Kinect センサーを用いる手法が提案されている[4]. Kinect で検出した頭部座標, 顔の 3 次元回転角, 顔の 3 次元平行移動量, 左右目の 2 次元座標の計 17 個を特徴量にし, 連続した 5 フレームのデータを 1 つの受講者の動きとして, 5 フレームのデータがすべて欠損値の時には, 特定の行動を取っていたものとする. そうでない時は, 各特徴量の平均と分散を算出し, 特徴ベクトルとして用いている. 結果として, 1 つの行動に対して 34 次元のベクトルができている. このベクトルに対してクラス分類を行うために k 近傍法を用いて状態を推定する.

本研究では, 音声対話システムにおいて, 目の前にいる人たちに呼びかけ, システム側から積極的に対話できるようにするために, 彼らの行動が, こちらに接近しているのか, 離れて行っているのか, 単に通り過ぎているのかを識別する方法を検討する. 具体的には, Kinect センサーを用い, 人物の骨格検出を行い, 頭部の位置情報を取得する. そして, この位置情報から速度や傾きを求めてその人の接近や離脱を判定する.

ただし, 音声対話システム前でのユーザーの自然な動作を収集し, 曖昧な行動を判定することは, 最初の段階では大変困難と思われるため, 本論文では, 比較的単純化された歩行パターンに限定して検討を行う. 具体的には, 4 つの動作, (1) 接近, (2) 離脱, (3) 停止, (4) 通過, のみを対象とする. 被験者にはおよそのパターンを示し, それらに合わせるよう歩いてもらう. そのとき, Kinect センサーから得られる座標や距離を用い, 速度やその方向を利用して, 被験者の動作を識別する.

本論文の構成を示す. 2 節にて, 提案法として, 人物の接近・離脱・停止・通過に対する識別方法を述べる. 次に, 3 節では, 評価データの収録方法や実験方法, 適合率, 再現率, F 値による評価結果について述べ, 最後に, 4 節で本論文をまとめる.

## 2. 音声対話システム前での動作識別方法

### 2.1. 動作の定義

人物の動作推定として, まず, 簡単な歩行パターンのみを識別することを考え, ここで定義する歩行パターンのみを対象とする. ユーザーの動作として, (1) 接近, (2) 離脱, (3) 停止, (4) 通過の 4 種類を定義する.

まず, 図 1 に, 本研究で使用する上から見た時の Kinect センサーの$xz$座標系を示す. 図中にはないが, 紙面奥から手前方向が$y$軸正の向きになる. $y$軸は高さを表す. ただし, 本研究では高さの情報を扱わないため, $x$軸と$z$軸のみを処理の対象とする. また, Kinect から下へ向かって左右に開いている線分は, Kinect により深度距離を推定できる範囲を示す. 公表されている深度センサーのスペックは水平方向に 70 度とされている.

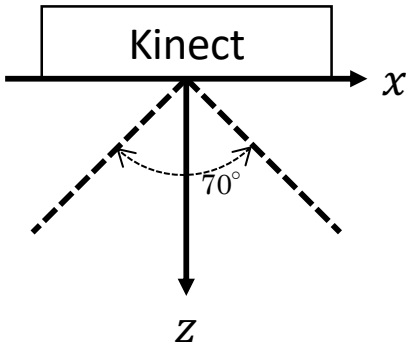


図 1　Kinect 座標系

定義した各動作を下記で説明する.

**[接近]**

図 2 に「接近」の歩行パターンを示す. Kinect（ここに音声対話システムがあると仮定）の方へ, (a) 正面から, (b) 左側から, (c) 右側から近づく歩行である.

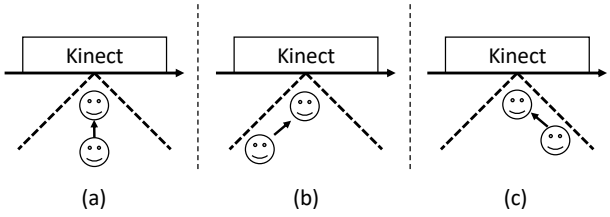


図 2　接近動作

**[離脱]**

図 3 に「離脱」の歩行パターンを示す.「接近」とは逆の方向へ歩くパターンで, Kinect から, (a) 正面へ, (b) 左側へ, (c) 右側へ遠ざかる歩行である.

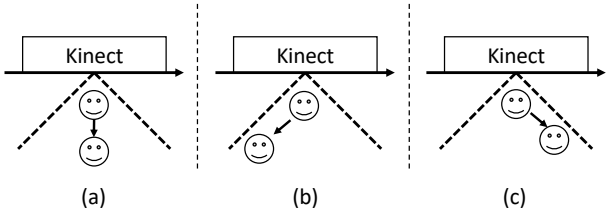


図 3　離脱動作

**［停止］**

歩行の動作をしていない状態とする．

**［通過］**

図 4 に「通過」の歩行パターンを示す．Kinect を，(a) 左から右へ，(b) 右から左へ通り過ぎる歩行である．

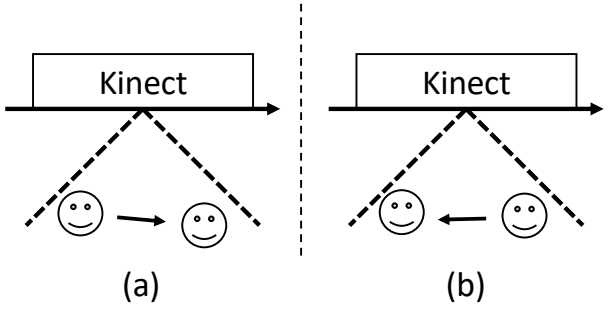


図 4　通過動作

## 2.2.　動作推定方法

図 5 に今回用いる動作推定方法の流れを示す．詳細は，2.3，2.4 節に示すが，まず，大まかな流れをここで説明する．接近と離脱の判定をするために，Kinect により，頭部の座標値から距離を求めて速度を算出し，歩行しているかどうかの判定をする．まず，欠損値の修正を行った後，距離・速度を求める．欠損値の修正は 2.4 節，手順 2 で述べるように，前後の平均値を使っている．次に，得られた速度が閾値を超えた時に「歩行」と判定するが，前後 3 点での値をスムージングして最終判定とする．歩いている時には，頭部の座標値から，歩行している動線の傾きを求める．Kinect（音声対話システム）に対する平行線に対しての動線の傾きが，ある一定の範囲内であれば，「通過」と判定する．そうでない時は，Kinect に近づいているか，離れているかのどちらかであるので，速度の符号でどちらか判定をする．符号がマイナスの時は「接近」，プラスの時は「離脱」と判定する．

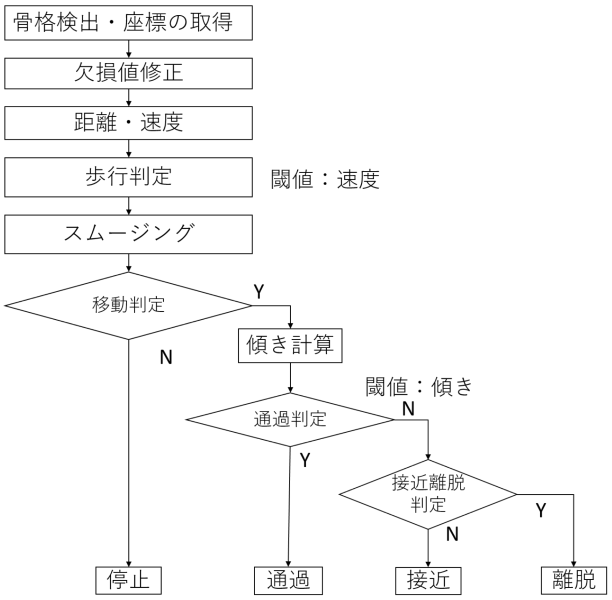


図 5　判定処理の手順

## 2.3.　距離と速度の計算

距離を推定する方法として 2 種類検討する．**[提案法 A]**では，Kinect 深度値から時刻$t$と$t+1$での原点からの距離を求め，距離の変動から接近／離脱を推定，速度は 2 点間の距離との大小から求めた符号を付与する．**[提案法 B]**では，Kinect の深度値を直接使用し，速度を推定する．以下に詳細を述べる．

**[提案法 A]**

図 6 の直線 A の距離を座標から計算する．まず，時刻$t$での座標と，時刻$t+1$での座標における Kinect（原点）からの距離$d_t$を求める．

$$d_t = \sqrt{x_t^2 + z_t^2}$$

次に，求めた距離の前後を比較する．このときの符号$f(d_t, d_{t+1})$を接近，離脱に対して決める．現時刻$t$での距離の方が大きい時は，近づいているのでマイナスとし，小さい時は，離れているのでプラスとしている．

$$f(d_t, d_{t+1}) = \begin{cases} 1, & d_{t+1} \geq d_t \quad \text{（離脱）} \\ -1, & d_{t+1} < d_t \quad \text{（接近）} \end{cases}$$

次に，2 点の座標の差を計算してから，三平方の定理により斜面を求める．

$$D_{t+1} = \sqrt{(x_{t+1} - x_t)^2 + (z_{t+1} - z_t)^2}$$

Kinect は 1 秒間に 6 個の頭部座標を取得することができ

る．この取得間隔を$T = 1/6$ [s]とする．次式で速度$\mathrm{v}_{t+1}$を求めることができる．

$$\mathrm{v}_{t+1} = \frac{d_{t+1} - d_t}{T}$$

**[提案法 B]**

図 6 の 2 つの直線 B の距離$\left\{d_t^{(\mathrm{B})}, d_{t+1}^{(\mathrm{B})}\right\}$を Kinect の深度値から直接取得し，利用する方法である．提案法 A での速度の定義を下記のように変更し，利用する．

$$\mathrm{v}_{t+1} = \frac{d_{t+1}^{(\mathrm{B})} - d_t^{(\mathrm{B})}}{T}$$

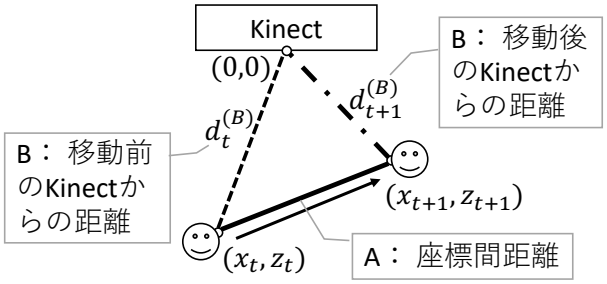


図 6　距離計算

## 2.4.　推定アルゴリズム全体

前処理や詳細も含め，歩行動作推定アルゴリズム全体を次にまとめる．

**[歩行動作推定アルゴリズム]**

**手順 1．骨格検出・座標の取得**

Kinect でユーザーの検出を行い，骨格認識をする．次に，頭部を追跡し，頭部座標を取得する．

**手順 2．欠損値修正**

現在の座標$\boldsymbol{x}_t = (x_t, z_t)$が欠損している場合で，かつ，前後時刻の座標が 0 でない時（値が欠損していない時），前後時刻の座標平均$\bar{\boldsymbol{x}}_t$を用いる．

$$\bar{\boldsymbol{x}}_t = \frac{\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t+1}}{2}$$

**手順 3．距離$d$・速度$v$**

速度は提案法 A，B の方法でそれぞれ推定する．頭部座標$\boldsymbol{x}_t$，または，手順 2 の平均座標$\bar{\boldsymbol{x}}_t$を用い，速度を求める．

**手順 4．歩行判定**

手順 3 で得られた速度が速度の閾値を超えているかどうかを次の式$f(v_t)$で判定する．速度の閾値を超えていれば 1，そうでない時は 0 とする．

$$f(v_t) = \begin{cases} 0, & |v_1| \geq |v_t| \\ 1, & |v_1| < |v_t| \end{cases}$$

**手順 5．スムージング**

手順 4 の結果をスムージングするために，以下のように三項移動平均$\bar{f}_t$を計算する．

$$\bar{f}_t = \frac{f(v_{t-1}) + f(v_t) + f(v_{t+1})}{3}$$

**手順 6．移動判定**

手順 5 の結果を次式で判定する．1 未満の時は「停止」，1 の時は「歩行」とする．

$$f(\bar{v}_t) = \begin{cases} 0, & \bar{f}_t < 1 \quad \text{（停止）} \\ 1, & \bar{f}_t = 1 \quad \text{（歩行）} \end{cases}$$

**手順 7．傾き計算**

最小二乗法を用い，傾きを計算する．前後 3 点からもっとも近い直線の傾き$s_t$を計算し，それを傾きとして用いる．

$$s_t = \frac{\sum_{i=-1}^{1}(x_{t+i} - \bar{x})(z_{t+i} - \bar{z})}{\sum_{i=-1}^{1}(x_{t+i} - \bar{x})^2}$$

**手順 8．通過判定**

図 7 に示すように，移動する動線の傾きがある範囲内にある時に「通過」，そうでない時，「接近」または「離脱」と判定する．傾き$s_t$が傾きの閾値$\alpha$を超えている場合，0 として次の処理に進む．そうでない時は$-2$として「通過」と判定する．

$$g(s_t) = \begin{cases} -2, & |s| < |a| \quad \text{（通過）} \\ 0, & |s| \geq |a| \quad \text{（接近または離脱）} \end{cases}$$

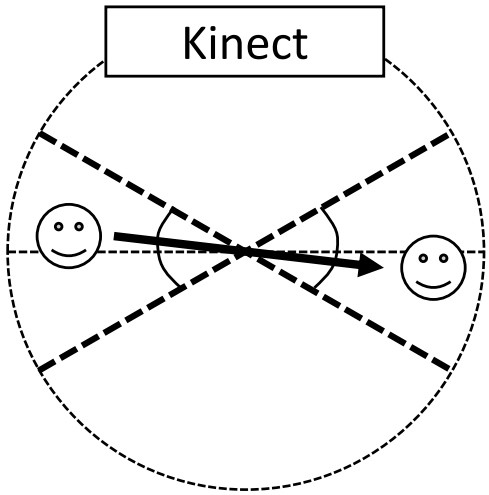


図 7　通過判定範囲

**手順 9. 接近・離脱判定**

ここで定義している速度は負の時，Kinect に「接近」，正の時に「離脱」となる．下記のように値を定義する．

$$h(v_t) = \begin{cases} 1, & v_t > 0 \quad (\text{離脱}) \\ -1, & v_t < 0 \quad (\text{接近}) \end{cases}$$

# 3. 歩行動作の識別実験

## 3.1. 評価データ

本研究で用いるデータの収録方法や条件について述べる．Kinect の前で，あらかじめ決めたルートを被験者に歩行してもらい，データを収録した．図 8 に収録環境を示す．著者らの大学研究室内で，Kinect が検出可能な前方 70 度内，3 m×3 m の範囲内において，被験者に歩行してもらった．Kinect は床からの高さを 160 cm とした．今回の実験では，簡単にするために，歩行ルートがほぼ一定になるよう，床に目印をつけた．図 9 に歩行ルートを，表 1 に被験者に歩いてもらうルートの種類を示す．被験者には最初におよそのルートや範囲を示し，後は自分のペースでそれぞれのルートを歩いてもらった．

表 2 に被験者や収録した評価データの情報を示す．被験者は 21 歳か 22 歳の男子大学生 20 名で，あらかじめ決めたルートを一人 10 回程度，歩行してもらった．人物検出プログラムは Microsoft 社 Kinect for Windows SDK 2.0，NtKinect [5]を用いて作成した．1/6 秒ごとに 1 フレーム取得した．正解ラベル付けをするにあたり，撮影時の動画を見ながら主観的に動作のフラグと「データ無し」フラグを用いて，この中のフラグ 1 つをフレーム 1 つ 1 つに割り当てた．

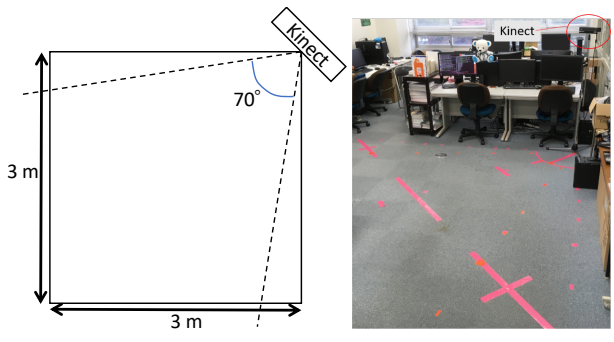


図 8 収録環境

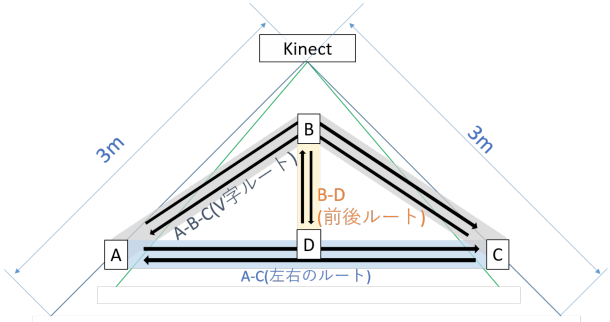


図 9 歩行ルート

表 1 歩行ルートの種類

| 前後のルート | 左右のルート | V 字のルート |
|---|---|---|
| B → D | A → C | A → B → C |
| D → B | C → A | C → B → A |

表 2 評価に用いたデータ

| 被験者 | |
|---|---|
| 被験者数 | 20 名 |
| 年齢層 | 20 代 |
| データ数 | 617 個 |
| 歩行ルート | 3 種類 |
| 各ルートのデータ数 | |
| 前後のルート | 232 個 |
| 左右のルート | 181 個 |
| V 字のルート | 204 個 |

## 3.2. 評価方法

4 つの歩行動作の識別を適合率，再現率，F 値で評価する．それぞれの定義を示す．

**適合率**：

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

**再現率**：

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

**F 値**：

$$\text{F} - \text{measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} \cdot \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

ここで，式中の記号説明を以下に示し，それらの関係を表

3 に示す.

TP (True Positive) :

正解と予想した結果が合っていた時の評価

FP (False Positive) :

正解と予想した結果が間違いだった時の評価

FN (False Negative) :

不正解と予想した結果が合っていた時の評価

TN (True Negative) :

正解と予想した結果が間違っていた時の評価

表 3　評価尺度内の記号定義

| | | 正解 | |
|---|---|---|---|
| | | 正 | 負 |
| 予測結果 | 正 | TP | FP |
| | 負 | FN | TN |

### 3.3.　速度の閾値に対する精度

図 10 に速度の閾値を変化させたときの再現率，適合率を示す．また，表 4 に，「停止」の F 値が最も高い時の停止の閾値，再現率，適合率と「停止」，「接近」，「離脱」の F 値を示す．速度の閾値では，「接近」と「離脱」をしている時の歩行を判定するために Kinect へ「接近」，「離脱」のしている前後と V 字ルートのデータを用いている．A，B は提案法を表し，A が座標間の距離を用いたもの，B が Kinect までの距離を用いたものである．

速度の閾値の結果は， 2 つの提案法であまり変化がない．また，表 4 の F 値の各数値をみてもほとんど同様な値であることが分かる．「接近」と「離脱」においては約 9 割弱検出できた結果となった．

表 4　「停止」の F 値が最も高い時の「停止」，「接近」，「離脱」の F 値

| | 閾値 | 停止 | | | F 値 | |
|---|---|---|---|---|---|---|
| | | 再現率 | 適合率 | F 値 | 接近 | 離脱 |
| A | 0.26 | 0.926 | 0.778 | 0.846 | 0.897 | 0.887 |
| B | 0.18 | 0.915 | 0.774 | 0.838 | 0.894 | 0.882 |

### 3.4.　傾きの閾値に対する精度

傾きの閾値は，通り過ぎている場合と接近離脱をしている場合を区別できるようにするため，3 つのルートのデータを用いている．

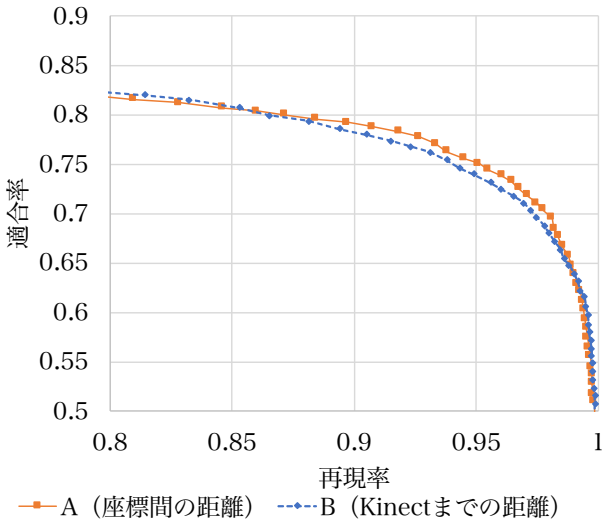


図 10　速度閾値を変化させた時の再現率，適合率

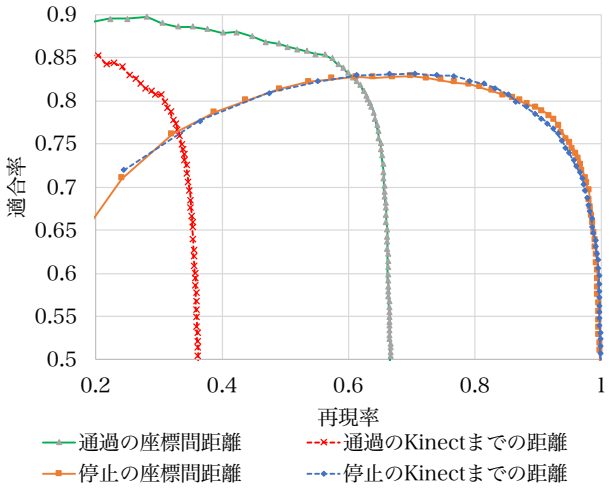


図 11　傾き閾値を変化させた時の再現率，適合率

求めた速度の閾値を用いて，傾きの閾値を変化させたときの再現率，適合率と図 10 を合わせたものを図 11 に示す．また，表 5 に， 速度の閾値と「通過」の F 値が最も高い時の閾値から得られる再現率，適合率を，表 6 に，「通過」の F 値が最も高い時の各動作状態の F 値を示す．このグラフから，「通過」での F 値は，「停止」での F 値よりも低いことがわかる．また，この表の F 値の「通過」は，図 11 からも分かるように A よりも B の方が低く，値としては 24%低くなっている．この原因は，再現率が低くなっているからである．再現率が低いのは，B の方法だと Kinect までの距離を用いており，横に移動した際に距離の変化量が少ない．そのため，その後の速度の計算にお

いて距離の誤差が小さく，「停止」という判定になっているからである．A の方法の場合には，座標間の距離を用いており，どの方向に歩いても歩行した分の距離を計算することができる．そのため，B よりも「通過」の再現率が高く，F 値も高くなる．

表 5　速度の閾値と「通過」の F 値が最も高い時の閾値から得られる再現率，適合率

| | 閾値 | | 通過 | |
|---|---|---|---|---|
| | 速度 | 傾き | 再現率 | 適合率 |
| A | 0.26 | 0.36 | 0.637 | 0.793 |
| B | 0.18 | 0.36 | 0.343 | 0.725 |

表 6　「通過」の F 値が最も高い時の各状態の F 値

| | F 値 | | | |
|---|---|---|---|---|
| | 停止 | 接近 | 離脱 | 通過 |
| A | 0.788 | 0.876 | 0.841 | 0.706 |
| B | 0.731 | 0.876 | 0.841 | 0.466 |

## 4. まとめ

本研究では，音声対話システムに接近したり，離れていく人に対し，呼び込みや挨拶をシステムが自律的に行えるようにすることを目標とし，画像や深度センサーを用いて人を検出し，その動作の識別方法を検討した．今回は Kinect を用い，画像や深度センサーの情報から，人が近づいてきたり（接近），離れていたり（離脱）する動作を識別する手法を提案した．具体的には，頭部座標から距離と速度を計算し，歩行判定をする．その後，歩く方向の傾きを求め，通り過ぎているか，接近または離脱かの判定をする．通り過ぎていなければ，速度の符号から接近であるか，離脱であるかを識別する．ここで用いる距離は，座標間の距離か，深度センサによる Kinect までの距離を用いて，それぞれの手法を評価をした．

将来的には，ユーザの実際の動作を収集して研究を行なっていきたいが，今回は初の試みであったので，歩行時の意図を明確にできるよう，行動パターンやルートを限定して行なった．行動パターンを「接近」「離脱」「停止」「通過」の 4 種類のみとし，あらかじめ決めた 3 つのルートを Kinect の前で被験者に歩行してもらい，被験者の座標や距離を収録した．

提案法として，Kinect からの距離を求め，その距離変動から推定する提案法 A と，Kinect で得られる深度値を直接使用して用いる提案法 B を提案し，評価した．これらの評価方法として，適合率，再現率，F 値を用いた．提案法にある閾値 2 つに対して最適な値を求めた．

その結果，「接近」，「離脱」を約 8 割識別することができ，どちらの距離を用いても変わらないことが確認できた．しかし，「通過」歩行には，座標間の距離が約 7 割検出し，提案法 B では，約 4 割しか検出できなかった．Kinect からの深度値をそのまま用いた時に F 値が低い原因は，横に移動した時に，一つ前の場所との距離の誤差が小さく，「停止」という誤判定になっていることが多かった．「停止」の推定精度は，提案法 A である座標間の距離を用いる方が提案法 B の Kinect からの深度値を用いる方法より約 5%高く推定できていた．今後，通り過ぎる人を検出する必要がある場合には，座標間の距離を用いる方が適切であることが分かった．

今後の課題として，まず，ラベルづけの問題がある．正解ラベル付けの際に，U ターンした際の動きなど細かい動きに対して，「停止」として正解ラベル付けを行っていた．これは行動パターンを 4 種類に限定していたためである．より自然な動きを推定するには，記述する行動パターンを増す必要がある．正解ラベルを増やして，詳細な動作を扱える手法に改善することで性能向上を図る必要がある．また，今回の評価データは簡略化した歩行パターンのみを対象としたが，音声対話システムを実際に用意して，自然な人の動きを収録，それらを分析，分類して，より詳しい動作の識別を検討していく必要がある．さらに，Kinect の機能として複数のユーザーを検出し，区別することができるが，今回は，このユーザーを区別する処理を行っていないため，提案法では 1 人しか対応できない．動作認識も複数人に対応できるような処理を考えていく必要がある．

## 謝辞

本研究の一部は JSPS 科研費 26330211 の助成を受けた．

# 参考文献


[1] 西村 竜一, 西原 洋平, 鶴身 玲典, 李 晃伸, 猿渡 洋, 鹿野 清宏, “実環境研究プラットホームとしての音声情報案内システムの運用,” 電子情報通信学会論文誌. D-II, 情報・システム, vol. 87, no. 3, pp. 789-798, 2004.

[2] 中井 慎, “音声対話エージェントによる愛知工科大学の案内システム,” 愛知工科大学卒業論文, 2009.

[3] 有満 大輝, “Kinect を用いた頭部追跡によるリアルタイム音声強調の研究,” 大分大学卒業論文, 2013.

[4] 神園 卓也, 高野 茂, 馬場 謙介, 村上 和彰, “深度情報を含む映像からの行動認識に関する研究,” 火の国情報シンポジウム 2014, 3B-3, 2014.

[5] 新田 善久, “NtKinect : C++ Class Library for Kinect V2,” 情報処理学会ヒューマンコンピュータインタラクション(HCI)研究会, 2017-HCI-172, pp. 1–6, 2017.