

3次元仮想空間とのインタラクションに適した ジェスチャインタフェース・ツールの開発

村田龍吾, 富田勝義, Le Van Nghia,
木村朝子, 大島登志一

本研究では、ユーザを取り囲む仮想空間との3次元的なインタラクションにおいて、複数の身体部位の位置や姿勢の状態によってユーザの全身動作をジェスチャインタフェースとして利用する。そこで、アプリケーション・システムにジェスチャインタフェースの機能を容易に組み込むことを目的として、ジェスチャの設計支援や登録機能を備えたソフトウェアツールを開発する。本稿ではジェスチャ認識手法とツールの機能について述べる。

Development of a Gestural Interface Tool for Interaction with 3D Virtual Space

Ryugo Murata, Katsuyoshi Tomita, Le Van Nghia,
Asako Kimura and Toshikazu Ohshima

This paper focuses on a gestural user interface which uses full-body motion interacting with 3d virtual space surrounding a user. We assume to use positions and orientations of multiple points on a user's body than just a few acceleration data, for deciding the gestures. For that purpose, we are developing a software tool for aiding implementation of this kind of gesture interface into VR/MR application systems. Essential functions of the tool include assisting for gesture design, capturing and analysing motion data, recognising gestures, and interactive testing. The gesture recognition approach and the functions of the tool are described in this paper.

1. はじめに

近年、様々なデジタル電子機器の普及などによりその入力方式も多様化してきている。その中でもユーザの身体動作をシステムとの対話操作に用いるジェスチャインタフェースが注目されてきている。身近な例では、スマートフォンやタブレット型端末のタッチパネルディスプレイによる指先の直接的な操作は、すでに一般的なものとなっている。また、家庭用ビデオゲームでも、任天堂社のWiiリモコン、Microsoft社のKinectやソニーコンピュータエンタテインメント社のPS Moveなど、従来のゲームパッドのようなボタン入力ではなく、ユーザの身体動作を操作に使用するスタイルが主要なインタフェースの一つとして定着しつつある。

ジェスチャインタフェースは、直感的で自然な対話操作が可能であるという特長から前述のように利用が進んでいるが、ジェスチャインタフェースをアプリケーション・システムに実際に組み込むには、まずジェスチャを設計し、そのジェスチャを認識するための機構を実装することとなる。その実装過程においては、センサの計測値と動作との対応関係を十分に把握した上で、実現したいところのジェスチャの同定に適した特徴量の選択と認識アルゴリズムおよび判定条件の設計を行う等、使用するセンサ・計測手段とジェスチャの種類に依存した経験的な作業を伴うという課題がある。

自由度が高々数軸程度の加速度データなどであれば単純なパターン認識モデルも比較的容易に適用可能ではあるが、特にVR (Virtual Reality) やMR (Mixed Reality) において、ユーザを取り囲む3次元空間全体を操作対象とした、全身の動作を伴うジェスチャインタフェースの場合には、上記のような課題はより顕著となる。そこで本研究では、ユーザの複数の部位の位置・姿勢情報を一次情報として利用することを前提としながらも、Wiiリモコンなどの加速度センサやKinectのような距離画像式モーションキャプチャなども包含し得る汎用性の高い枠組みを検討し、ジェスチャインタフェースをシステムに実装する上での前記のような作業を支援するソフトウェアツールの開発を行っている。

本ツールは、身体動作の計測と記録、動作の特徴分析の支援、ジェスチャテンプレートの登録、ジェスチャ認識機能などを提供する。アプリケーション側からは、本ツールでのジェスチャ認識結果を通信するためのモジュールを組み込むことで、その機能を利用することができる。本稿では、ジェスチャインタフェースをアプリケーションに組み込むワークフローの分析を行い、その作業を支援するツールとして必要な機能について述べ、本目的のために開発したジェスチャ認識手法とソフトウェアツールについて説明する。

2. 関連研究

ジェスチャインタフェースは様々な分野・目的で利用されている。そのためジェスチャに関する研究も様々なアプローチが行われており、Grandhiらは対象に触れて行うジェスチャではなく空間上で行うパントマイムのようなジェスチャの設計方針に関する分類を行い、ジェスチャ設計のガイドラインを作成した[1]。Ashbrookらが開発したデザインツール“MAGIC”[2]では日常動作のデータを収集し、そのデータを分析することで普段の生活の中で行われる動作によってジェスチャの誤認識が発生しにくいジェスチャの設計を支援している。またHartmannらの“Exemplar”[3]ではセンサを用いたインタラクションにおいて、センサから得られる入力データとアプリケーションを関連付けるためのオーサリングツールを開発し、設計者がジェスチャ設計において必要となる設計、テスト、分析の繰り返し作業を迅速に行えるためのアプローチとして、ジェスチャを実演することによって閾値と入力ジェスチャの関係をグラフィカルに表示し、その結果から直感的に適切な閾値の分析や変更を実現し作業の効率化を支援している。

これらの従来研究では計測部位は一箇所のみであるが、本研究では6自由度の位置姿勢情報を取得可能なセンサによって同時に複数点を計測する構成において、それらの相対的な位置関係や時間的な変化など様々な特徴量から全身を使った多様な動作をジェスチャとして設計・認識可能なツールを開発する。上記の研究の知見を踏まえた上で、ジェスチャ設計・登録において考慮すべきファクターが増加した場合における作業工程の変化や問題点、技術的課題を試験的にツールの運用を繰り返すことで洗い出す。

3. ジェスチャ認識

3.1 身体動作の特徴量

本研究では複数の身体部位を計測することにより、計測部位の位置や姿勢の時間的な変化からジェスチャを認識する。また本研究で想定するジェスチャはユーザを取り囲む周囲の3次元空間に対してインタラクションを行う事を想定しているため、ジェスチャ認識に用いる特徴量は動作者の向きや位置に依存することなく不変であることが望まれる。そこで図1に示すように特徴量はあるセンサを基準とした座標系の値に変換することによって不変性を確保する。この変換には計測部位の位置ベクトル \mathbf{P}_i と姿勢行列 \mathbf{O}_i をユーザが選択した基準となる計測部位の位置ベクトル \mathbf{P}_o と姿勢行列 \mathbf{O}_o から式(1)により相対的な位置ベクトル \mathbf{P}'_i 、姿勢行列 \mathbf{O}'_i に変換することにより行う。

$$\begin{bmatrix} \mathbf{O}'_i & \mathbf{P}'_i \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{O}_o^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{O}_i & \mathbf{P}_i - \mathbf{P}_o \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

しかし、ジェスチャ認識の問題点として誤認識の発生がある。ジェスチャの誤認識を防ぐためには複数の特徴量を用いることによって識別することが考えられる。そこで1種類の特徴量では識別が困難となる類似動作を識別可能にするために、位置・姿勢の時間的な変化から取得可能な速度、加速度、角速度などを適宜選択して利用する。

3.2 提案するジェスチャ認識モデル

本研究におけるツールに対する要求として以下の項目が挙げられる。

- ・ ジェスチャ設計・登録の容易性
- ・ 設計者の意図が反映可能な自由度の高い設計の実現
- ・ 実時間でのジェスチャ認識処理

実時間によるジェスチャ認識の実現のために、連続的に入力される時系列のデータの中から登録されているテンプレートと類似するパターンを識別する手法として、ジェスチャ認識にも利用される連続DPマッチングを利用する[4]。連続DPマッチングでは時間長や速さの異なる2つの時系列データに対して類似度を計算することが可能であるためジェスチャ認識に適しているが、ジェスチャの開始・終了時点を検出する必要がある。そこで図2に示すように状態機械を併用し、連続DPマッチングを状態の中の1つに組み込むことによって前後の状態によりジェスチャの開始条件・終了条件を設定し、ジェスチャの実行区間の検出を行う。また各状態に複数の遷移条件を設定することにより、設計者の理想とするジェスチャインタフェースを実現する。

以上のことからジェスチャは複数の状態から構成され、各状態に設定された条件を満たすことにより、条件に応じた先の状態へ遷移を行い、設定した最後の状態まで遷移を行った際にジェスチャを認識成功と判断する。本研究で開発するツールでは遷移条件を大きく分けて以下の2つの種類を用意し、2つの遷移条件のどちらか、もしくは2つを組み合わせることで状態遷移モデルを構築することでジェスチャ設計・登録を行う。

(1) 条件判定

センサから取得した特徴パラメータの現在のフレームにおける状態（速度の大きさ、計測部位の距離等）が設定した閾値を上回る、もしくは下回るなどの判定により状態遷移を行う遷移条件。

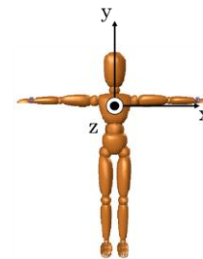


図1 特徴空間座標系

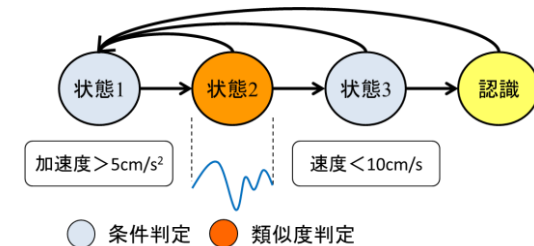


図2 認識モデルイメージ図

(2) 類似度判定

類似度判定はテンプレートマッチングを用いている状態の遷移条件であり、連続 DP マッチングにより算出した 2 つのパターン間の類似度が、2 つのジェスチャが類似していると判断するための閾値より高い場合に次の状態へ遷移を行なう条件。

3.3 連続 DP マッチング

本ツールで使用する連続 DP マッチングの処理を説明する。本ツールでは条件判定によりジェスチャの開始時点を検出し、その時点における入力データからジェスチャテンプレートとの連続 DP マッチングを開始する。連続 DP マッチングでは毎フレーム更新される入力データに対してテンプレートのデータとの類似度を動的計画法により入力フレーム i におけるテンプレートのフレーム j に対する相違度 $D(i,j)$ を式 (2) を用いて算出する。ただし $d(i,j)$ は入力フレーム i とテンプレートの参照フレーム j の局所距離とする。局所距離 $d(i,j)$ は式 (3) からユークリッド距離を算出する。

$$D(i, j) = \min \begin{cases} D(i-1, j) + d(i, j) & \text{①} \\ D(i, j-1) + d(i, j) & \text{②} \\ D(i-1, j-1) + d(i, j) & \text{③} \end{cases} \quad (2)$$

$$d(i, j) = \sqrt{(R_{ix} - I_{ix})^2 + (R_{iy} - I_{iy})^2 + (R_{iz} - I_{iz})^2} \quad (3)$$

この漸化式から得られる図 3 に示すワーピングパスの $D(t,T)$ における値を入力とテンプレートの類似度として利用する。また現在の入力フレームにおけるテンプレートとの最適な対応付けを行うフレーム (対応フレーム f) を式 (4) を用いて現在のフレームにおける類似度の正規化した値 $D'(t,j)$ から求める。

$$f = \arg \min_j D'(t, j) = \frac{D(t, j)}{t + j} \quad (4)$$

ここで得られた $D'(t,j)$ が最小となる j 番目のフレームを対応フレームとして判断し、図 4 に示すように対応フレームがテンプレートのフレーム数に対して占める割合からジェスチャ入力シーケンスの終了を判断する。また連続 DP マッチングの高速化のため整合窓を設定し、ワーピングパスを対応フレームから W フレーム先までに限定することで極端な対応付けを回避することができ、図 3 に示す赤色の領域のように最適なパスから外れた不要だと考えられるパスを除外することで計算量を削減している。

4. ジェスチャツール

4.1 概要

本ツールはアプリケーション・システムにジェスチャインタフェースの機能を容易

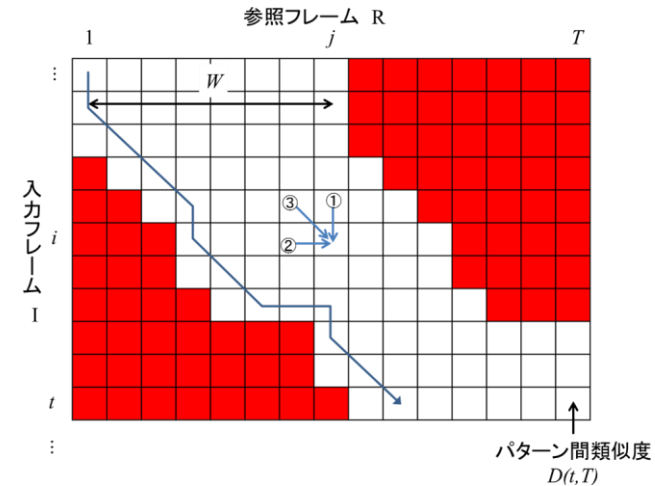


図 3 連続 DP による類似度計算

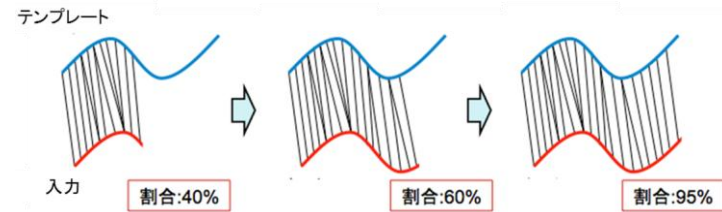


図 4 終了判定

に組み込むことを目的としたツールであり、そのためのジェスチャ設計・登録支援機能とジェスチャ認識機能を備える。ジェスチャ設計・登録支援機能では情報の視覚化、直感的な入力方法、設定の自動化といった観点から支援を行い、ジェスチャ認識ではリアルタイム性、ジェスチャ設計の自由度といった点を考慮し、3 章で述べた認識モデルを用いて認識を行う。ツール使用の基本的な流れとしては以下のようにしており、ツールの構成図を図 5 に示す。

4.2 計測機能

本ツールではジェスチャを設計・登録するために、ジェスチャインタフェースとして利用したい動作を実際に行い、センサを用いて動作を計測することにより動作のテンプレートを取得する。この動作の計測には動作に適したセンサを用いて行う。

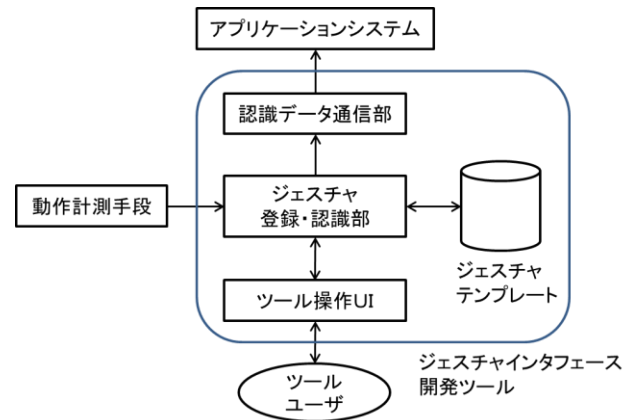


図5 ツール構成概要図

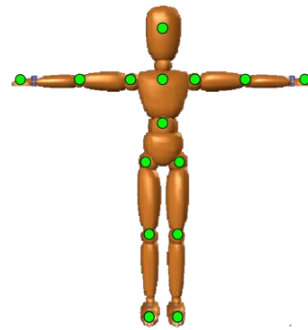


図6 対象となる計測部位

【利用を想定する主な位置姿勢計測手段】

(1) 磁気式センサ

磁気センサは、遮蔽に強くマークが隠れるといった心配を必要がない。しかし、磁気を発生させるトランスミッタから離れることにより位置推定精度が低下する。また、磁気センサは有線である場合が多く、その場合センサを装着することにより身体的な拘束が発生するなどの問題がある。

(2) 光学式センサ

光学式センサは、複数のカメラとマークを用いて3次元的な位置姿勢を計測する方式であり、比較的広範囲の動作計測が可能であるが、マークが複数のカメラから観測不可能になると計測が出来ず、また、身体に複数のマークを装着する必要があるといった欠点もある。

(3) Kinect センサ

Kinect センサは、距離画像情報と機械学習による身体姿勢推定によって、身体にマーク等を装着する必要なく身体部位の計測が可能という利点があるが、更新速度が遅く、遮蔽物や自身の体で計測部位が Kinect から観測不可能になるとトラッキングが破綻するといった精度の面での欠点がある。

【計測部位】

動作の計測をするに当たって全身の動きをジェスチャに利用するため、一般的にアニメーションなどでも定義されている部位として図6に示す15点を計測対象となる身体部位として定義し、計測の際にはどのセンサと計測部位が対応するかを選択する。このように計測部位をフォーマット化することによりデータベースの統一を図る。

4.3 登録機能

本ツールによるジェスチャ認識を行うために、ジェスチャ認識モデルを構成する動作のテンプレート、状態数、遷移条件を設定することによりジェスチャを設計し、データベースへジェスチャを登録する。しかし、これらの設定をツールのユーザが手動で行うにはジェスチャ設計の経験が必要となる。そこで動作の分析の支援、遷移条件の自動設定を可能にすることによりユーザの負担を軽減する。

【動作の分析】

計測機能で取得した動作データをグラフ化し、動作計測時に撮影した動作映像と同期して表示することにより視覚的に計測値と動作の対応関係をユーザに分かるようにすることによってユーザの理解を支援する。図7に示すように情報提示ウィンドウではウィンドウを4分割し左上に身体動作映像、特徴ベクトルのx軸を右上、y軸を左下、z軸を右下に表示する。

【ジェスチャ設計・登録】

ジェスチャ設計・登録は以下の流れに沿って行う。

(1) ジェスチャテンプレートの抽出

分析結果から動作の特徴が表れている区間をグラフ中からマウスのドラッグにより範囲選択を行うことによってジェスチャ実行領域として抽出し、ジェスチャのテンプレートとして用いる。

(2) 状態の分割

(1)で抽出した区間内の状態を分割したいフレームをマウスで右クリックすることにより、そのフレームを状態が遷移するフレームとして設定し、ジェスチャを複数の状態に分割することができる。ポーズジェスチャなどの1つの状態でジェスチャが構成される場合にはこの作業は必要ない。

(3) 遷移条件の設定

認識モデルに基づいて、各状態における遷移条件を計測データの分析結果からどの特徴量・計測部位を用いて行うかを設定する。設定した図9に示すダイアログに表示され、状態1で設定した遷移条件がジェスチャの開始を判定する条件となる。

【ジェスチャ自動設計】

ジェスチャの自動設計を行う際の前提条件として、設計されるジェスチャは2つの状態から構成され、本ツールでは一次情報として位置を使用することから、状態1の開始条件の判定にはテンプレートの初期フレームにおける位置の情報からポーズを判定し、ポーズの類似度を用いて状態遷移を行う。状態2ではテンプレートの位置情報を用いた連続DPマッチングによる類似度判定を行うジェスチャが設計される。

ジェスチャの自動設計では同一のジェスチャとするテンプレートを複数取得し、一つのクラスタとしてまとめる。クラスタに所属するM個のテンプレート間の相違度の総和が最小となるテンプレートを式(5)を用いてクラスタを代表するテンプレ

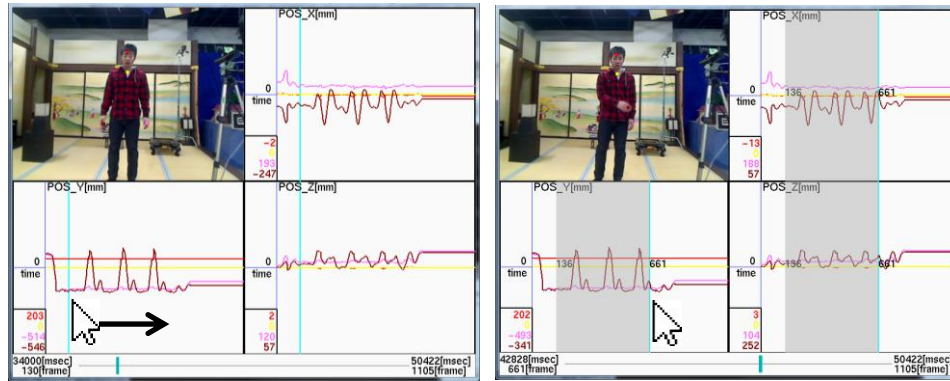


図 7 テンプレート抽出

トとして選出する。代表テンプレートを用いて同一クラス内のテンプレートがすべて認識可能となる値を式 (6) を用いて算出し、閾値として設定する (図 9)。

これによりジェスチャ設計者は登録したい複数の同様の動作を計測することでジェスチャの登録を行うことが可能となる。

$$t = \arg \min_i \sum_{j=1}^M Dist(T_i, T_j) \quad (5)$$

$$threshold = \max Dist(T_t, T_i) \quad (6)$$

【認識テスト】

設計したジェスチャが設計者の意図した通りのインタフェースを実現できているとは限らない。ジェスチャ認識では行った動作に対して表 1 に示すような結果が起こり得るため、遷移条件等のジェスチャ設計の調整を行うことによってジェスチャとして認識すべき動作を認識し、認識すべきでないジェスチャを排除する必要がある。

そこで認識テストではジェスチャデータベースから複数のジェスチャを選択し、選択したジェスチャの中から調整を行うジェスチャを選択する。その後、選択したジェスチャとして認識すべき動作を実行する。その結果認識できなかった場合にはそのジェスチャが認識可能となる遷移条件へと自動的に更新し、実行した動作が認識成功した場合には遷移条件の更新は行わないため動作を入力として繰り返すことにより、自動的に閾値が調整される。また調整するジェスチャと選択したジェスチャの認識経過を同時に表示することにより、誤認識が発生する可能性のあるジェスチャや他のジェスチャとの関連性を俯瞰することができる。



図 8 ジェスチャ設計ダイアログ

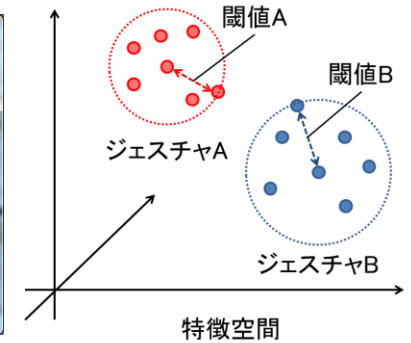


図 9 閾値の自動計算

表 1 認識結果テーブル

結果 出力	True	False
Positive	実行動作が意図したジェスチャとして認識 (True Positive; TP)	実行動作が意図しない別のジェスチャとして認識 (False Positive; FP)
Negative	ジェスチャではない動作を認識しない (True Negative; TN)	実行動作がジェスチャとして認識されない (False Negative; FN)

4.4 認識機能

【ジェスチャセット作成】

アプリケーション・システムで使用が想定されるジェスチャの集合 (ジェスチャセット) をデータベースの中の中から選択して作成する。またジェスチャを選択する際には各ジェスチャに認識成功時にアプリケーション・システムに出力する ID を付与する。この ID はジェスチャ毎に固有である必要はなく、必要に応じて同じ ID を複数のジェスチャに割り当てることによって異なるジェスチャを一つのグループとしてまとめることが可能になる。これによりアプリケーション・システムでは受けとった ID に対する処理を記述するだけでジェスチャインタフェースを組み込むことができる。

【結果出力】

本ツールでは提案するジェスチャ認識モデルを用いて実時間での時系列データの中からスポッティング認識を行う。しかしスポッティングの結果複数のジェスチャが現在実行中と判断している場合、どのジェスチャが最適解かを判断する必要があるが、候補となる全てのジェスチャが終了してから判断するのでは動作終了から結果を返すまでに遅延が発生する。そこで本ツールでは図 10 と図 11 に示す出力結果とするジェスチャを選ぶための選択規則を 2 種類用意した。

(1) 早期認識ジェスチャ優先選択

スポッティングを行った結果、複数のジェスチャが現在実行中と判断され、その中のあるジェスチャが終了したと判断した際には最初に認識成功と判断し終了したジェスチャが結果として選択し、結果として選択されたジェスチャ以外の実行中と判断されているジェスチャは認識を中断する。

(2) 類似度優先選択

類似度判定をジェスチャ認識に用いている場合には、ジェスチャの終了時に入力動作と実行候補ジェスチャの類似度が求まっているため、あるジェスチャが終了判断した際に他の実行候補ジェスチャの類似度を比較し、最も類似度が高いジェスチャを結果として出力する。ジェスチャの類似度は $D(t, T)$ を用いて判断するため、ジェスチャが実行の冒頭段階では極端な対応付けや初期化による低い類似度となる。そのため終了したあるジェスチャが終了した時点よりも類似度が高い他のジェスチャが存在した場合には、入力動作は類似度の高いジェスチャが実行されたと判断し優先する。

5. 運用

ツールの問題点を洗い出すためにツールを用いたジェスチャ設計・認識実験を行った。実験では被験者をジェスチャテンプレートとする動作群と認識実験を行う動作群に分け、テンプレートには動作を行う人物本人のテンプレートが存在しない条件の下を行った。また認識実験を行う動作群では 10 名の被験者を集め、ジェスチャ試行前には行う動作を提示して見せた後に各ジェスチャを 10 回試行した結果から TP, FP, FN の値を算出する。今回の実験ではジェスチャの設計はツールの開発者が設計した。

5.1 実験 1

【実験概要】

実験 1 では、ジェスチャインタフェースで頻繁に利用される手を使ったジェスチャを図 12 に示す 4 パターン用意したジェスチャセットに対して認識実験を行った。実験 1 のジェスチャのテンプレートは 2 名の被験者からテンプレートを作成している。このジェスチャセットではジェスチャ開始・終了位置、動作軌跡などに複数の共通する動作をジェスチャ間に持たせることによって認識結果に与える影響を分析する。

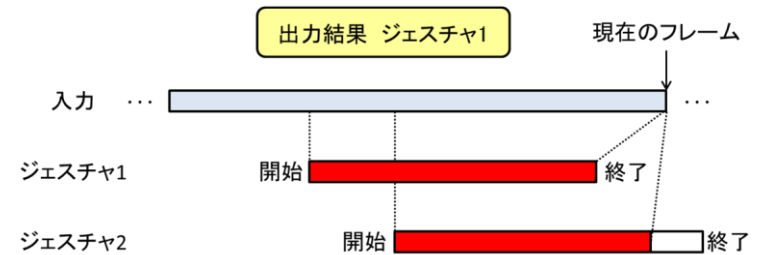


図 10 早期認識ジェスチャ優先結果

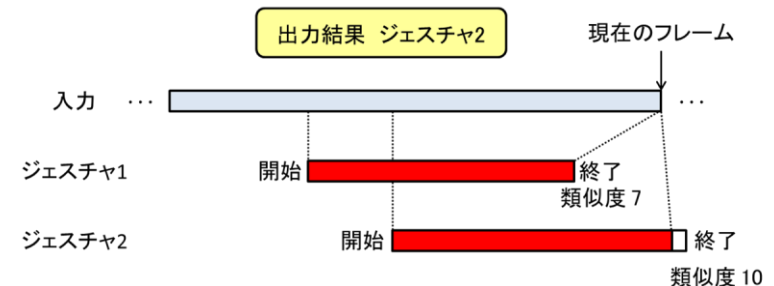


図 11 類似度優先結果

【結果】

実験 1 の被験者毎の認識精度とジェスチャごとの認識精度は図 13 に示す通りである。いずれの被験者も高い精度で認識に成功しており、被験者自身のテンプレートがないにもかかわらず 100% 認識に成功している被験者も存在する。全体の認識精度の平均も 90% を超えている。

5.2 実験 2

【実験概要】

実験 2 では、エンタテインメント性のある動作として、図 14 に示すチャンバラゲームを想定した際の刀を振る動作を 5 パターン用意したジェスチャセットに対して認識実験を行った。実験 2 では 3 名の動作から被験者からテンプレートを作成している。

【結果】

実験 1 の被験者毎の認識精度とジェスチャごとの認識精度は図 15 に示す通りである。結果として上から下に振り下ろす動作の認識精度が一番高く、動作の軌跡が水平方向に近づくにつれ精度が低下していくという結果が得られた。一番認識精度が高かった被験者で 92%、低かった被験者で 60% と大きく差が開いた結果となった。

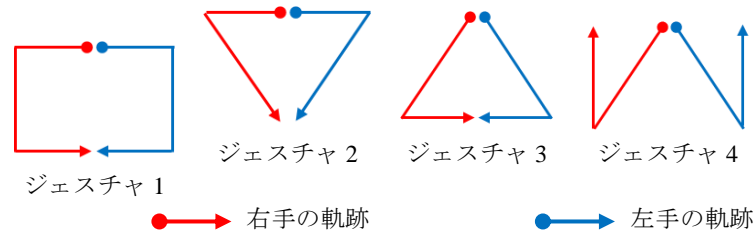
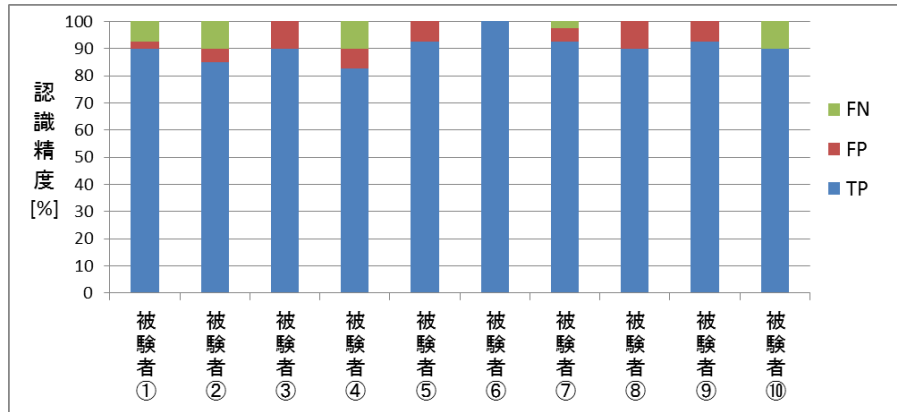
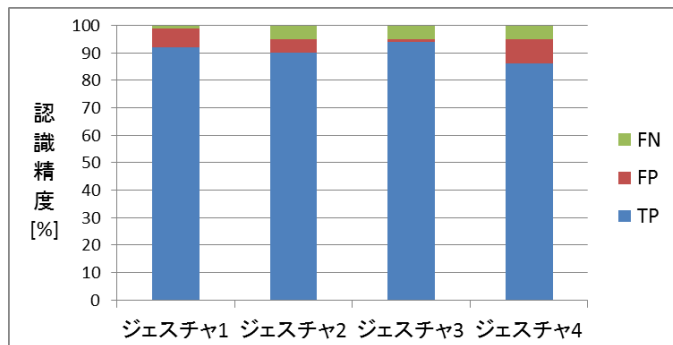


図 12 ハンドジェスチャセット



(a) 被験者別認識精度



(b) ジェスチャ別認識精度

図 13 実験 1 の結果

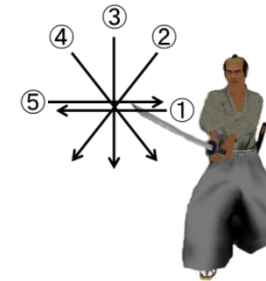
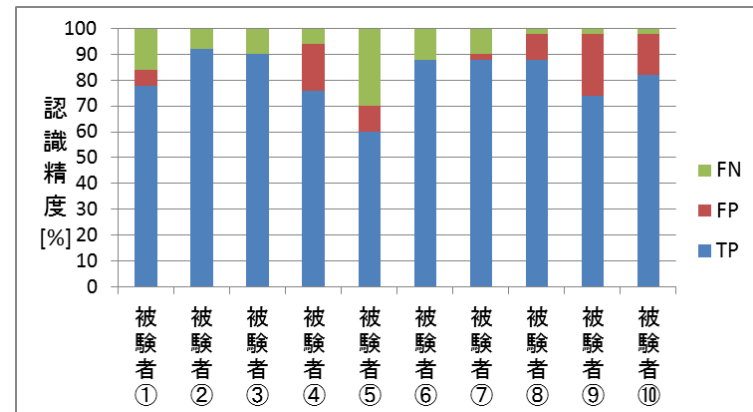
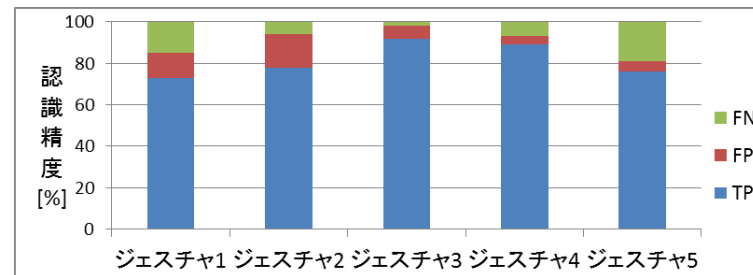


図 14 チャンバラジェスチャセット



(a) 被験者別認識精度



(b) ジェスチャ別認識精度

図 15 実験 2 の結果

5.3 考察

実験1と実験2の結果から、本人のテンプレートが存在せず、少ない数のテンプレート数でも実験1のように手のみを動かすような簡単で再現性の高いジェスチャであれば高精度で認識可能であることが確認できた。しかし、実験2のように体全体を動かす必要のある動作の場合にはより動作の個人差が生まれ認識精度が低下する。その一つの要因として特徴量計算における座標変換が考えられる。今回のツールでは人間の動作に着目するため座標変換を行うことによって、1方向に対しての動作のテンプレートを取得することで全方位の動作に対応することが可能となっているため、複数方向のテンプレートを用意する必要がない。しかし、実験2のジェスチャで動作の軌跡が横に近づくにつれ認識精度が低下する傾向から、テンプレートとなる動作を行った人物には見られなかった腰の回転が影響し、座標変換による動作の個人差による誤差が大きくなっているためだと考えられる。上から下に振り下ろす動作には腰の回転が発生しない為、座標変換の影響が少なく左右の動きに比べ、比較的高精度な結果となったと考えられる。これは動作中に回転が加わる複雑なジェスチャではより顕著になると考えられるため、本研究で対象とする動作を考慮すると、この問題を解決し、より高精度な認識を実現する必要がある。また今回の実験では少ない数のテンプレートで行ったため、このような結果が得られたが、高精度な認識を行うためにテンプレートにはどれほどのバリエーションが必要なのかを検討する必要がある。

むすび

本研究では、アプリケーション・システムにジェスチャインタフェースを容易に組み込むことを目的とし、ユーザを取り囲む3次元空間全体を操作対象とした、全身の動作を伴うジェスチャインタフェースを設計する際の設計支援機能と認識機構を備えた、ジェスチャデザインツールを開発した。

まず、ジェスチャの構造について検討を行い、その結果から、連続DPマッチングによるテンプレートマッチングと状態機械の併用による状態遷移を用いた認識モデルを提案した。認識モデルにおける詳細な処理の内容として、本ツールで使用する認識アルゴリズムを説明した。

ジェスチャの設計過程ではセンサを用いて複数点を計測することによって得られる動作の時系列データを視覚化し、ジェスチャを分析するため支援とした。そして分析結果からツールを用いて必要な設定を行うことによりプログラミングを必要とせず動作の実演からジェスチャが設計できることを説明した。その後、2種類の実験を通してツールを運用し、ジェスチャの認識が行えることを確認した。

今後の展望として、認識アルゴリズムの改良による認識精度の向上、およびジェスチャ設計・登録の効率化が考えられ、高速化の方向性として認識を行うために有効で

あるとされる計測部位や特徴量の自動選択やテンプレートの自動抽出などの自動化が考えられる。またアプリケーション・システムで利用されるジェスチャとしては多様な動作であることを想定するため、ツールでは設計可能なジェスチャに自由度を持たせた結果、設計者に複数の入力を要求する。ジェスチャ設計者の意図通りの動作を行うためには、設計者の意思の介入は必要となるが、設計の容易化の観点からはこの問題について解決する必要があるため、今後多くのツールの試用実験の知見を重ね、そこから得られる評価から実際の運用における新たな課題を洗い出し、実用性の高いジェスチャツールの実現を目指す。

参考文献

- 1) Sukeshini A. Grandhi, Gina Joue and Irene Mittelberg: "Understanding Naturalness and Intuitiveness in Gesture Production: Insights for Touchless Gestural Interfaces," Proc. CHI, pp. 821 - 824, 2011.
- 2) Daniel Ashbrook and Thad Starner: "MAGIC: A Motion Gesture Design Tool," Proc. CHI, pp. 2159 - 2168, 2010.
- 3) Björn Hartmann, Leith Abdulla, Manas Mittal, and Scott R. Klemmer: "Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition," Proc. CHI, pp. 145 - 154, 2007.
- 4) 森明慧, 内田誠一, 倉爪亮, 谷口倫一郎, 長谷川勉, 迫江博昭: "ジェスチャの早期認識・予測ならびにそれらの高精度化のためのネットワークモデルに関する検討", 日本ロボット学会誌, Vol. 24, No. 8, pp. 954 - 963, 2006.