

# 加速度センサを用いたジェスチャ操作の早期認識手法の提案

出田 怜<sup>1</sup> 村尾 和哉<sup>1</sup> 寺田 努<sup>1,2</sup> 塚本 昌彦<sup>1</sup>

**概要:** 端末に搭載された加速度センサから得られる値を用いて端末の動きを検出することで従来のインタフェースでは困難であった入力を実現できる. 一般的に加速度センサを用いたジェスチャ認識では, ジェスチャ終了後に認識処理を開始するため, ジェスチャ入力を行ってからフィードバックの出力までに遅延が生じ, インタフェースの操作性の低下につながる. 本研究では, 加速度センサを用いたジェスチャ認識において, 途中までのジェスチャ入力データと教師データとの距離を逐次的に計算する早期認識アルゴリズムを提案する. 提案手法では各教師データとの距離計算において, 認識結果の候補が拮抗している場合には, 認識結果を出力せずに次サンプルの入力を待ち, 確度の高い唯一の候補が決定した時点で認識結果を出力する. タブレット端末使用時を想定した 27 種類のジェスチャに対して提案手法の性能を評価したところ, 提案手法はジェスチャの終了約 1 秒前に 100%の精度で認識できることを確認した. また, 提案手法を組み込んだジェスチャベースのフォトビューアを実装した. 提案手法を用いることでジェスチャを早期に認識でき, 操作性が向上することを確認した.

## Early Gesture Recognition Method with Accelerometers

RYO IZUTA<sup>1</sup> KAZUYA MURAO<sup>1</sup> TSUTOMU TERADA<sup>1,2</sup> MASAHICO TSUKAMOTO<sup>1</sup>

### 1. はじめに

マイクロエレクトロニクスの発展によるコンピュータの小型化や高性能化, 軽量化に伴い, 携帯電話やタブレットパソコンといったコンピュータを持ち歩いて, いつでも利用する形態であるモバイルコンピューティングが人々の生活に浸透している. しかしながら, デバイスの小型化によるボタンの減少や表示領域の縮小によって操作性は低下しているといえる. これに対して, 端末に加速度センサなどの小型センサを内蔵し, それらを応用することでより操作性を向上させたデバイスが数多く登場しており, その応用が注目されている. 特に, iPhone や Android 端末などのスマートフォンや Wii や PS3 などの家庭用ゲーム機のコントローラには加速度センサが内蔵され, センサの値から端末の傾きや動きなどを検出して, ブラウジング中のスクロール操作や 3D オブジェクトの描画, キャラクタの操作などが直観的に行える. このことから今後, 加速度センサ

によるジェスチャ認識を利用した入力インタフェースが数多く登場することが予想される.

一般的に加速度センサを用いたジェスチャ認識では, ジェスチャ終了後に認識処理を開始するため, 例えば, 端末を傾けて 1 秒後に画面内の表示が回転するといったように, インタフェースの操作性の低下につながる. 単純にジェスチャの途中で認識するアプローチでは, 途中までの動作が類似している複数のジェスチャを学習している場合, 認識精度の低下を招く恐れがある. そこで本研究では, 未知のジェスチャデータとあらかじめ登録しておいたジェスチャデータとの距離計算を逐次的に行い, あるジェスチャの尤度が他のジェスチャの尤度に対して相対的に高くなった時点で認識結果を出力することで, 認識精度を低下させることなく早期にジェスチャを認識する手法を提案する.

以降, 2 章で関連研究を紹介し, 3 章ではジェスチャを早期に認識するための提案手法について述べる. 4 章で評価実験を行い, 5 章では提案手法を用いたアプリケーションを紹介する. 最後に 6 章で本研究をまとめる.

<sup>1</sup> 神戸大学大学院工学研究科  
Graduate School of Engineering, Kobe University  
<sup>2</sup> 科学技術振興機構さきがけ  
PRESTO, Japan Science and Technology Agency

## 2. 関連研究

### 2.1 ジェスチャ認識を用いたアプリケーション

富林らは両手に装着した加速度センサによりジェスチャを認識し、DJ パフォーマンスを行う DJ 支援システムを提案している [1]. 数種類のジェスチャを音量の調整や、曲の再生、停止などに割り当て、認識結果に応じて操作を行い、見た目にもインパクトがあるパフォーマンスを実現している. Liu らは Nokia 研究所が提唱する直線や円など 8 種類の図形を描く動作を 1 個の 3 軸加速度センサで認識するシステムを提案している [2]. また、提案システムを用いて、ビデオ共有サービスにおけるビデオリストのモバイル端末向け閲覧インタフェースを実装している. Agrawal らは携帯電話に搭載された加速度センサの値から空間に描いた文字を認識するシステムを提案している [3]. 空間に文字を書く際に、文字を書くスピード等に制約を設けることによって 83% の認識精度を達成している. 矢谷らはモバイル端末における情報の移動を直観的に実現する Toss-It というインタフェースを提案している [4]. Personal Digital Assistant (PDA) に加速度センサを取り付けてトスの投げ上げの角度や速度、振りの回転方向などを検出し、カメラを用いてユーザ同士の位置を測定することによって、特定のユーザへ情報を投げ渡すといった直観的な操作でデータ送信ができる.

しかしながら、これらの認識システムはジェスチャ終了後に認識処理を開始しており、認識結果に応じたフィードバックの出力に遅延が発生し、インタフェースの操作性が低下すると考えられる. 出力の遅延を軽減するためには、ジェスチャ終了前に認識処理を開始しなくてはならない.

### 2.2 ジェスチャの早期認識

菅家らは実際に存在するドラム (実ドラム) と実際には存在しないドラム (仮想ドラム) の叩打の識別をスティックに取り付けた加速度センサを用いて行い、仮想ドラム叩打時のみ音を出力する Airstic Drum を提案している [5]. 適切なドラム演奏を行うためには叩打と同時に音を出力する必要があり、直前に叩打の識別を行わなければならない. そこで叩打直前の特徴的なデータ波形に着目し、閾値を設けることによって叩打の識別を行なっている. しかし、多数の動作に対応するためには個別に閾値を設定する必要がある. また、このシステムでは叩打といった比較的単純なジェスチャを対象としているが、より複雑なジェスチャを対象とすると認識は困難であると考えられる. 藤本らはダンスステップを認識して BGM に合わせた効果音を再生する手法を提案している [6]. BGM の拍の前後の 2 段階でダンスステップの認識を行い、各段階で出力の修正を行うことで、認識精度および効果音の出力のタイミングの改善を

行なっている. 効果音を出力する場合は認識が間違っても即座に修正すれば聴衆への影響は小さいが、入力インタフェースにおいては認識結果に応じて出力したコマンドを取り消したり修正したりすることは困難である.

入力インタフェースにおけるジェスチャ認識では、認識結果を確定して出力するタイミングが重要となる. たとえば、認識対象となるジェスチャの種類が多いと認識結果が高い確度を得るまでに長い入力データが必要である. また、ジェスチャの途中段階まで類似したジェスチャが認識対象に含まれる場合は、類似した部分が終了するまで結果の出力を待たなければならない. 村尾らは、認識するジェスチャの種類を変化させた時の認識精度について実験から評価を行っており、類似したジェスチャを認識対象から除外しながらジェスチャ数を減少させるに従い、認識精度が上昇することを確認している [7].

このように、認識結果を出力するタイミングは認識対象とするジェスチャの種類および認識したいジェスチャの数に依存するため、高い精度で可能な限り早期に認識結果を出力するためにはそのタイミングを動的に変更する必要がある. 本研究では、入力波形と教師データとの距離を逐次的に計算し、最近傍の教師データと次点の教師データとの距離の差が相対的に大きくなった時点で認識結果の確度が高いと判断し、最近傍の教師データに付与されているジェスチャを認識結果として出力する手法を提案する.

## 3. 早期認識手法

### 3.1 システム構成

一般的に加速度センサを用いたジェスチャ認識では、認識したいジェスチャのデータとジェスチャのラベルのペア (教師データ) を事前に登録しておく必要がある. 登録したすべての教師データと認識したい未知のデータ (入力データ) との距離を比較し、距離が最も小さい教師データのラベルを入力データの認識結果として出力する. 一般的には、ジェスチャ終了までの入力データと教師データを用いて距離計算を行うが、本研究では図 1 に示すように、入力データが途中までであることを考慮した認識アルゴリズムを提案する. 提案する認識アルゴリズムでは 2 段階の計算処理から成る. 第一段階では、入力データとすべての教師データとの距離を計算し、距離の最も近いものと 2 番目に近いものを検索する. 第二段階では、最も近い距離と 2 番目に近い距離を比較することで認識結果の候補が拮抗している場合には、認識結果を出力せずに、より長い入力データを用いて判断し、確度の高い結果を出力する.

### 3.2 距離計算アルゴリズム

提案手法では、距離計算のアルゴリズムとして Dynamic Time Warping (DTW) [8] を用いる. DTW は時間的非線形圧縮距離を計算することで、長さや部分的に速度が異な

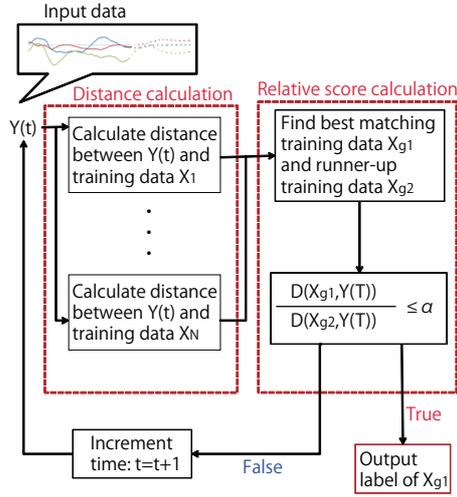


図 1 システム構成

Fig. 1 System structure.

るデータの距離を計算でき、ジェスチャ認識の距離計算において主要なアルゴリズムである。しかしながら、DTW は入力データと教師データがともにジェスチャの開始から終了までの完全なデータであることを想定しており、途中までの入力データに対して DTW をそのまま適用すると、入力したデータが完全なジェスチャであるとして処理されるため正しく認識できない。そのため、本研究では、途中までの入力データにも対応した距離計算を行うように DTW のアルゴリズムを改良する。

具体的なアルゴリズムを以下に示す。ある時点における入力データの長さが  $t$  であるとする、その入力データに合う教師データの部分波形の長さも  $t$  に近いと想定し、距離計算の対象と教師データの長さに制約を設け、 $t - \varepsilon$  から  $t + \varepsilon$  とする。この時の  $\varepsilon$  を検索幅と呼ぶ。教師データ  $X = (x_1, \dots, x_m)$  と長さ  $t$  の入力データ  $Y(t) = (y_1, \dots, y_t)$  の距離を計算する際、 $m \times t$  行列の  $d(x_i, y_j) = (x_i - y_j)^2$  を定義する。次に、 $X$  と  $Y(t)$  のインデックスのペアからなるワーピングパス  $F = (f_1, \dots, f_k)$  を検索する。ここで、ワーピングパスは次の 3 つの条件を満たす。

- 境界条件  
 $w_1 = (1, 1), w_k = (t + \varepsilon, t)$
  - 連続性  
 $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow a - a' \leq 1 \wedge b - b' \leq 1$
  - 単調性  
 $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow a - a' \geq 0 \wedge b - b' \geq 0$
- 上記の条件を満たす最小のパスを検索するために、以下の手順を適用する。

**Initialization:**

$$\begin{aligned} f(0, 0) &= 0 \\ f(i, 0) &= \infty \text{ for } i = 1, \dots, t + \varepsilon \\ f(0, j) &= \infty \text{ for } j = 1, \dots, t \end{aligned}$$

**Do for**  $i = 1, 2, \dots, t + \varepsilon$

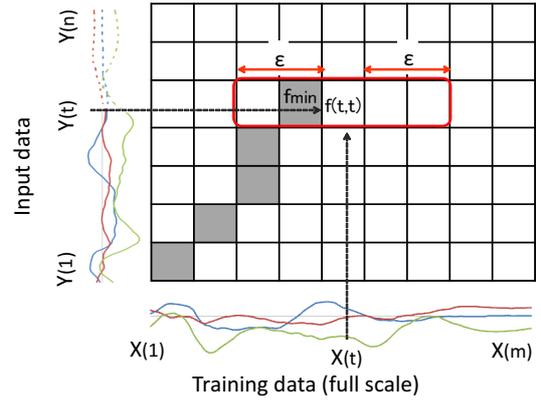


図 2 距離計算アルゴリズム

Fig. 2 Distance calculation algorithm.

**Do for**  $j = 1, 2, \dots, t$

$$f(i, j) = d(x_i, y_j) + \min \begin{cases} f(i-1, j-1) \\ f(i-1, j) \\ f(i, j-1) \end{cases} \quad (1)$$

**Output:**

Return  $D(X, Y(t)) =$

$$\min \begin{cases} f(t - \varepsilon, t) / (t - \varepsilon + t) \\ f(t - \varepsilon + 1, t) / (t - \varepsilon + 1 + t) \\ \vdots \\ f(t + \varepsilon - 1, t) / (t + \varepsilon - 1 + t) \\ f(t + \varepsilon, t) / (t + \varepsilon + t) \end{cases} \quad (2)$$

以上の手順によって、図 2 のような  $f$  の行列が作成される。これをタイムワーピング行列と呼ぶ。 $f(t - \varepsilon, t)$  から  $f(t + \varepsilon, t)$  までの値は時刻  $t$  における入力データと時刻  $t - \varepsilon$  から  $t + \varepsilon$  までのそれぞれの教師データとの距離を表している。図では、行列内の影のつけられた部分が最小のパスである。したがって、 $D(X, Y(t)) = f(t - 1, t)$  が最小の距離であり、時刻  $t$  における  $X$  と  $Y(t)$  との距離となる。検索幅を採用して  $t - \varepsilon \leq i \leq t + \varepsilon$  の中から最小値を検索することによって教師データとして登録したジェスチャのスピードと入力データのジェスチャのスピードが異なる場合でも適切な距離計算ができる。

この距離計算をすべての教師データについて行い、距離の最も近いジェスチャ  $g1$  と 2 番目に近いジェスチャ  $g2$  を導き出す。つまり、 $D(X_{g1}, Y(t))$  と  $D(X_{g2}, Y(t))$  を導き出す。ただし、 $X_{g1}$  はジェスチャ  $Y(t)$  と最も距離の近いジェスチャの教師データであり、 $X_{g2}$  はジェスチャ  $Y(t)$  と 2 番目に距離の近いジェスチャの教師データである。上記の計算過程を入力データが追加されるたびに行う。ただし、タイムワーピング行列の計算は追加された入力データに相当する部分のみを計算するだけであるため、入力データが追加されるたびに行う計算処理の負荷は小さい。

### 3.3 相対スコア計算アルゴリズム

距離計算アルゴリズムによって得られた  $D(X_{g_1}, Y(t))$  と  $D(X_{g_2}, Y(t))$  を用いて相対スコアを計算する. 相対スコアを用いることで入力データのジェスチャの冒頭部が複数のジェスチャと類似している場合, 結果の出力を猶予して, より長い入力データを用いて確度の高い結果を出力する. 具体的には, 時刻  $t$  における最小の距離  $D(X_{g_1}, Y(t))$  と 2 番目に小さい距離  $D(X_{g_2}, Y(t))$  の比が次式を満たすとき, 認識結果を出力する.

$$\frac{D(X_{g_1}, Y(t))}{D(X_{g_2}, Y(t))} < \alpha \quad (3)$$

本研究では式 (3) の左辺を相対スコアと呼ぶ.  $\alpha$  ( $0 < \alpha < 1$ ) は閾値であり, 閾値が小さいほど結果を出力するために 2 番目の距離との差が大きくなければならず, より確度の高い結果が得られることを意味する. 式 (3) を満たさなければ, 認識結果の出力を行わず, 入力データを  $Y(t+1)$  に更新し, 再度 3.2 節の距離計算を行う. また, 極端に短い入力データではジェスチャの判別が困難であるため, 入力データ数には下限を設定して, 入力データが 10 サンプル (0.2 秒) 以下では出力しない. 提案手法では認識結果の確度が高い時点で認識結果を出力するため, あらかじめ認識結果を出力する時刻を決めなくてもよいが, ジェスチャ終了まで待っても確度が高くない可能性がある. その場合は, ジェスチャ終了と同時に最も確度の高い認識結果を出力する.

ここで, 具体的な数値を用いた距離計算の例を図 3 に示す. 実際は加速度センサのデータは 3 軸であるが, ここでは簡単化のために 1 軸で説明する. ジェスチャ開始から終了までの教師データを  $X = \{5, 12, 6, 10, 6, 5, 18, 20, 10, 7\}$ , 与えられる入力データを  $Y(t) = \{11, 6, 9, 4, 2, \dots\}$  とする.  $t = 1$  から順に 1 サンプルずつ入力されるたびに入力データに対して図 3 に示すようにタイムワーピング行列  $f(i, j)$  を作成していく. ただし, ここでは入力データ数の下限は設定せずに説明する. まず,  $Y(1) = 11$  が入力されると, タイムワーピング行列の 1 行目が計算される. 検索幅を  $\epsilon = 1$  設定すると, この時刻における距離となる候補は  $f(1, 1)$  と  $f(2, 1)$  であり, 最小値である  $f(1, 1) = 6$  が時刻  $t = 1$  における  $X$  と  $Y(t)$  の距離となる. さらに,  $Y(2) = \{11, 6\}$  が入力されると, 2 行目のタイムワーピング行列が作成され, この時の距離の候補は  $f(1, 2)$  と  $f(2, 2)$  と  $f(3, 2)$  であり, その中で最小値である  $f(3, 2) = 7$  が時刻  $t = 2$  における距離となる.  $f(1, 2) = f(3, 2)$  のように, 候補の中で同じ値をもつ要素が存在する場合, 教師データのインデックスが大きい方を距離として選択する. このように, 入力データが更新されるたびにタイムワーピング行列を計算し, 相対スコアが閾値  $\alpha$  を下回ったときに認識結果を出力する. 相対スコアが閾値  $\alpha$  を下回らなければ認識結果を出力せずに, 入力データを更新して再度距離計算を行う.



図 3 距離計算の例

Fig. 3 Detailed example of distance calculation.

## 4. 評価

### 4.1 評価環境

タブレット端末使用時に行われると想定する 27 種類のジェスチャのデータ (表 1) を 3 軸加速度センサを 1 個配置した盤面を用いて 1 人の被験者から採取し, 認識精度と認識結果の出力からジェスチャ終了までの残り時間の関係を次節で述べる比較手法と提案手法で調査した. 各ジェスチャ 8 回ずつのデータを採取し, 1 回分を教師データ, 残りの 7 回分をテストデータとした. 使用した加速度センサはワイヤレステクノロジー社製 WAA-006[9] で, サンプル周波数は 50Hz でデータを採取した. 表 1 に各ジェスチャの平均動作時間を示す. 27 種類のジェスチャの動作時間の平均は 1.40 秒であり, 最短は 0.66 秒, 最長は 2.34 秒である. 被験者の解釈の違いによる影響を軽減するために, ジェスチャを被験者に教示する際は言葉や図ではなく, 実際の動作を示して行なった. センサデータは東芝社製 PC (dynabook, RX3 TM240E/3HD) に Bluetooth で送信され, PC 上で計算処理を行なった.

### 4.2 比較手法

早期認識の単純な手法として, ジェスチャの開始から一定時間までのデータ波形を用いて距離計算を行うことが考えられる. 具体的には, 長さ  $t$  のデータ波形が入力されたときに認識結果を出力するとし, 長さ  $t$  の教師データと距離計算を行い, 距離が最も小さい教師データのラベルを認識結果とする. この比較手法では, 長さの最も短い教師データに合わせて認識タイミングを決定しないといけない.

### 4.3 結果と考察

比較手法と提案手法それぞれにおいて, 入力データの長さを変えた時の, ジェスチャの残り時間と認識精度を図 4 に示す. ジェスチャの残り時間とは入力されるジェスチャの全長から入力されたジェスチャの長さを引いたものである. 議論を簡単にするため, 入力されてから結果が出力さ

表 1 ジェスチャー一覧  
Table 1 List of gestures.

ID	Description (Length[sec])	Illustration
1	Tilt to the near side (1.57)	
2	Tilt to the far side (1.43)	
3	Tilt to the left side (1.65)	
4	Tilt to the right side (1.73)	
5	Tap upper side twice (0.84)	
6	Tap left side twice (0.78)	
7	Swing twice to the left side quickly (1.36)	
8	Swing twice to the right side quickly (1.44)	
9	Shuffle cards (1.25)	
10	Tap lateral edge as though sifting (1.30)	
11	Scoop (1.84)	
12	Lay cards (1.57)	
13	Gather cards (1.60)	
14	Rap table with the longer lateral edge (1.23)	
15	Rap table with the surface of the board (1.28)	
16	Knock the board twice (0.88)	
17	Turn the board over (1.30)	
18	Rotate clockwise on the table (1.55)	
19	Shift up (1.33)	
20	Shift down (1.09)	
21	Shift left (1.29)	
22	Shift right (1.17)	
23	Shift diagonally up (1.13)	
24	Shift diagonally down (1.06)	
25	Draw a circle (1.75)	
26	Draw a triangle (1.88)	
27	Draw a square (2.25)	

れるまでの処理時間は考慮していない。計算処理時間については次節で述べる。比較手法では、入力データの長さを1サンプルずつ変化させて認識精度の計算を行なった。ま

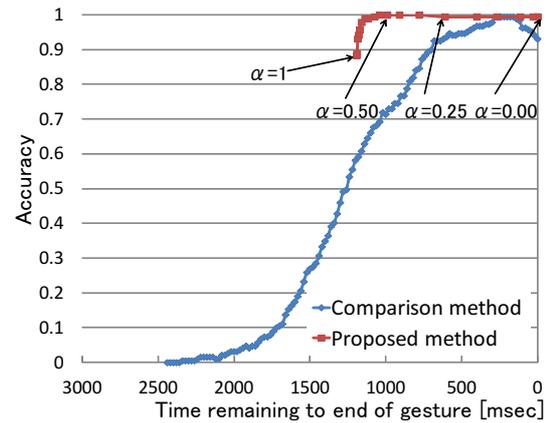


図 4 ジェスチャー終了までの残り時間に対する認識精度の変化  
Fig. 4 Accuracies vs. time remaining to the end of gesture.

た、教師データが入力データより短いとき、教師データの終端にセンサ値 0 を追加して長さをそろえた。提案手法では、閾値  $\alpha$  の値を 0 から 1 まで 0.05 ずつ変化させて認識精度の計算を行なった。図中の縦軸はジェスチャーの平均認識精度を表しており、横軸はジェスチャーの残り時間を表している。また、検索幅  $\varepsilon$  の大きさは 10 である。

図 4 より、比較手法において、ジェスチャー終了 800 ミリ秒前に平均認識精度が 0.8 を越えており、ほとんどすべてのジェスチャーがジェスチャー終了 800 ミリ秒前に認識できている。しかしながら、いくつかのジェスチャーでは低い認識精度を示した。特に認識精度の低かった 8 番、12 番、18 番、22 番、25 番、27 番のジェスチャーについて、認識精度とジェスチャーの残り時間の関係を図 5 から図 10 に示す。図中の縦軸はジェスチャーの認識精度、横軸はジェスチャーの残り時間を表している。図 5 に示す 8 番のジェスチャーでは、ジェスチャー開始から 800 ミリ秒間は認識精度が低く、この時、6 番のジェスチャーと誤認識していた。これは、6 番と 8 番のジェスチャーの冒頭部のデータ波形が類似しており誤認識したと考えられる。同様の理由で、22 番は 8 番と誤認識していた。さらに図 6 に示す 12 番のジェスチャーは 13 番のジェスチャーと誤認識していた。この 2 つのジェスチャーはそれぞれ逆方向の動作をするジェスチャーであるが、12 番と 13 番のジェスチャーの手を広げるまでの動作が類似してしまったと考えられる。図 7 に示す 18 番のジェスチャーは、ジェスチャー開始直後の認識精度が低い。この原因として 18 番のジェスチャーは同じ速度で動作することが難しく、データ波形の時間軸方向の歪みが大きかったため、比較手法のような単純に入力データと同じ長さの教師データと距離計算を行うアプローチでは適切に計算が行えなかったと考えられる。図 9 に示す 25 番のジェスチャーは、誤認識したジェスチャーに規則性がなく、動作の開始直後では認識結果の候補が拮抗していると考えられる。図 10 に示す 27 番のジェスチャーは、開始直後では 2 番や 24 番といった奥に移動さ

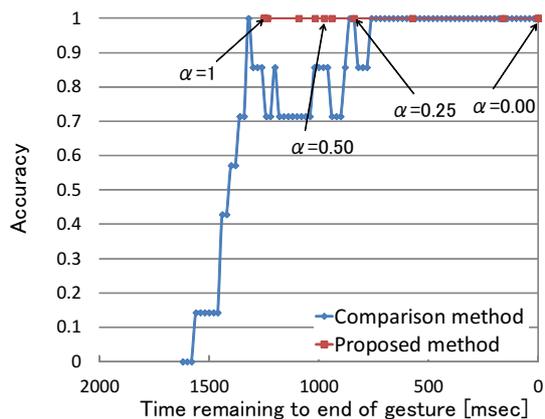


図 5 ジェスチャ 8: 右に 2 回振る  
**Fig. 5** Gesture 8: Swing twice to the right side quickly.

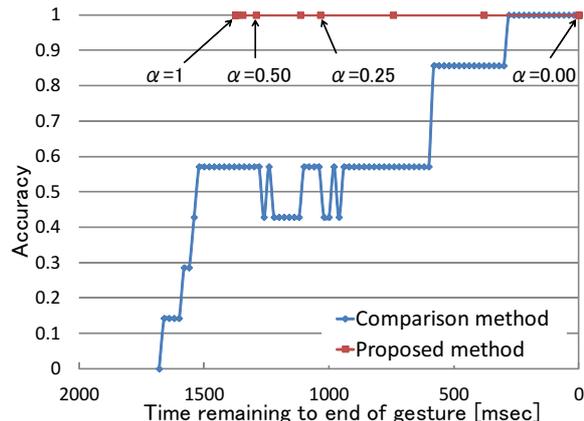


図 6 ジェスチャ 12: 机の上にトランプを広げる  
**Fig. 6** Gesture 12: Lay cards.

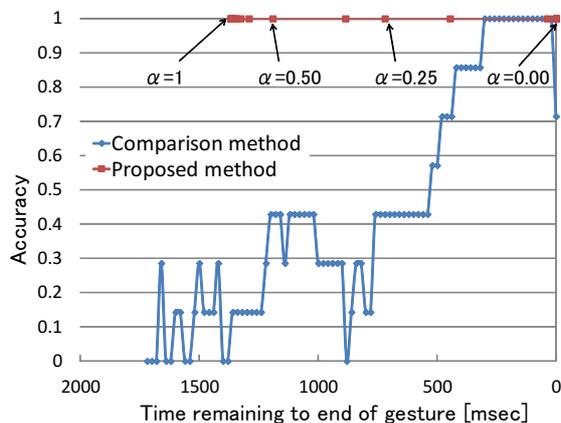


図 7 ジェスチャ 18: 机の上で時計回りに 1 回転させる  
**Fig. 7** Gesture 18: Rotate clockwise on the table.

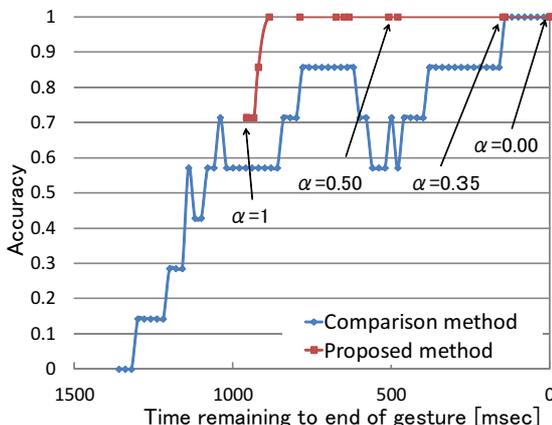


図 8 ジェスチャ 22: 右に平行移動させる  
**Fig. 8** Gesture 22: Shift right.

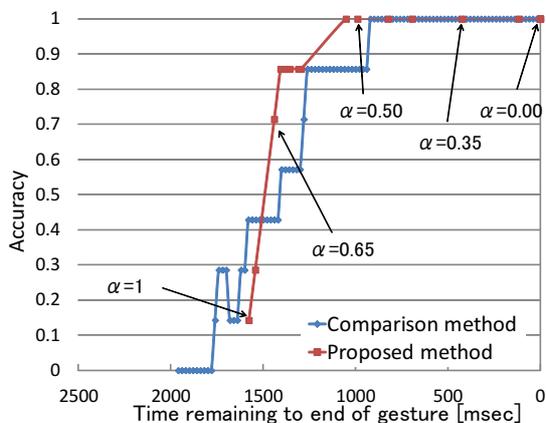


図 9 ジェスチャ 25: 丸を描く  
**Fig. 9** Gesture 25: Draw a circle.

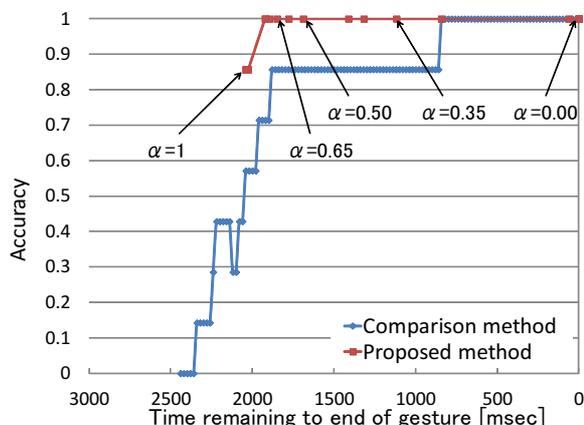


図 10 ジェスチャ 27: 四角を描く  
**Fig. 10** Gesture 27: Draw a square.

せるジェスチャと誤認識していた。27番のジェスチャは他のジェスチャを組み合わせで構成されたジェスチャであると考えることができ、高い精度を得るためにはある程度の長さの入力データが必要であると考えられる。認識精度が低いジェスチャもジェスチャ終了200ミリ秒前では高い認識精度で認識できていることがわかる。しかしながら、27番のジェスチャではこの時点で開始から約2秒経過しており、他の大部分のジェスチャは終了しているため、それらのジェスチャでは認識処理による遅延が生じると考えられる。

一方、提案手法において、図4より、閾値 $\alpha$ を0.45から1.00の間で設定するとジェスチャ終了の900ミリ秒前にジェスチャを認識し、高い精度で認識結果を出力できることがわかる。特に、 $\alpha$ が0.45, 0.50, 0.55のとき、認識精度が1.00を示しており、高い認識精度を維持しつつ出力の遅延を軽減している。 $\alpha$ が0から0.40のときは、高い認識精度を示しているが、残り時間が短くなっている。これは閾値 $\alpha$ の設定が厳しく、より長い入力データを必要としたためである。これに対して、 $\alpha$ が0.7から1.0のときは、認識精度が急激に落ちている。これは閾値 $\alpha$ の設定が緩く、精度の低い結果を出力したためである。閾値 $\alpha$ を低くすると、より早期に認識結果を出力できるが、入力データが短くなるため冒頭部が類似しているジェスチャ同士を認識対象としている場合、認識精度が下がってしまうと考えられる。

8番、12番、18番、22番、27番のジェスチャでは提案手法を用いることによって同じ残り時間でも比較手法よりも認識精度が向上している。これは検索幅 $\varepsilon$ を用いることによって適切な長さの教師データと距離計算ができていからであると考えられる。さらに提案手法では、閾値 $\alpha$ を適切に設定することで高い認識精度を維持しつつ、早期に認識ができる。しかしながら、提案手法を用いても早期に認識できなかったジェスチャが存在した。25番のジェスチャでは、ジェスチャ開始直後における認識精度の改善はみられなかった。25番のジェスチャは開始直後でさまざまな候補と拮抗していたため、高い認識精度で早期に認識するのは困難であったと考えられるが、適切な閾値を設定することによって精度は向上している。また、入力データが適切な長さになるまで出力の結果を待つことができ、より精度の高い認識結果を得ることができる。

以上の結果より、提案手法を用いることで最後までの入力データを用いた場合と同等の認識精度を維持しつつ、ジェスチャの終了1秒前に認識できることを確認した。

#### 4.4 処理時間に関する考察

図11に比較手法と提案手法における処理時間を示す。図の縦軸は認識に使用する入力データが入力されてから認識結果を出力するまでの処理時間、横軸は入力データの長

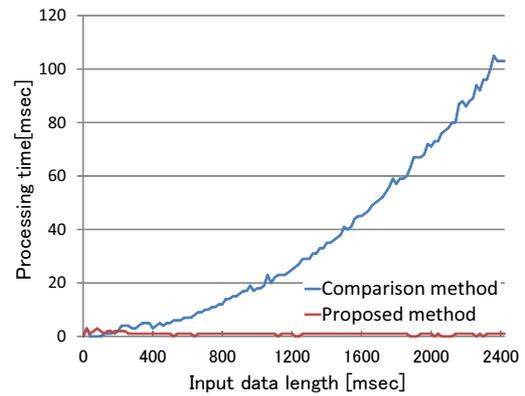


図 11 各入力データ長に対する処理時間

Fig. 11 Processing time for each input data length.

さを表している。図11より、提案手法の処理時間は常に一定で小さいことがわかる。これは、入力データが入力される度に逐次的に毎回同じ数のタイムワーピング行列を計算するためである。一方、比較手法の処理時間は入力データサイズの2乗に比例して増加していることがわかる。これは、認識を開始する時点で初めてタイムワーピング行列を計算するためである。提案手法はワーピング行列の計算に加えて、相対スコアの計算処理が加わるため、入力データ短いと比較手法の方が高速であるが、開始200ミリ秒以降は提案手法が逆転している。

比較手法であらかじめ決められた長さの入力データが与えられた時点で認識結果を出力するする場合を考える。長さが最も短いジェスチャである5番の0.78秒未満であり、図11より0.78秒の入力データは約10ミリ秒であるため、入力データ長が0.78秒で認識しなければならない。しかし、図5から図10の比較手法の結果より、ジェスチャ開始から0.77秒分のデータでは認識精度が低いことがわかる。一方、提案手法で最悪の場合を考えると、入力データがすべて与えられた時点でも相対スコアが低く結果を出力できない場合である。27番のジェスチャが最も長く、その長さは2.25秒であるが、その長さのデータの処理時間は図11より約1ミリ秒であるため、たとえ高い精度が得られずジェスチャ終了直前までのデータを必要とした場合でも、逐次的に認識処理が行われるため、出力の遅延はほぼ発生しないと考えられる。

## 5. アプリケーション

提案手法を用いたジェスチャ認識を利用して実装したアプリケーションを紹介する。図12に実装したアプリケーションのスクリーンショットを示す。実装したアプリケーションはスマートフォンでの画像ビューアであり、ジェスチャ動作に閲覧におけるいくつかの操作を割り当てている。例えば、デバイスを左右に傾けると表示されている画像を変更でき、手前に移動させるとズームができる。ま

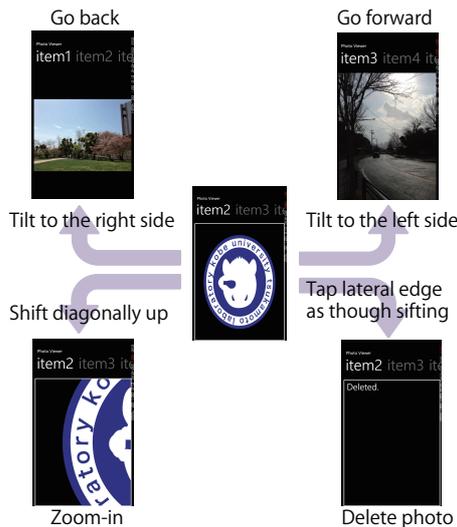


図 12 アプリケーション実装例

Fig. 12 A gesture-based photo viewer.

た、デバイスの左側面を叩くと表示している画像を削除できる。一般的なジェスチャ認識では、ジェスチャ終了後に認識処理を開始するため、ジェスチャと応答に時間差が生じて快適な操作性を実現することは困難であったが、提案手法を用いることでジェスチャの終了前に画面の応答を確認でき、ジェスチャにおける操作性の低下を軽減できる。

また、最近のスマートフォンは、ディスプレイサイズの大きいものが数多く登場し、両手で操作する場面が多く見受けられる。指の届かない位置のボタン操作など、片手操作を実現するためにはジェスチャ操作が有効であると考えられる。

他の応用例として、スポーツゲームやシューティングゲームなどが考えられる。これらはユーザの入力に即座に反応しなければならず、早期認識が有効であると考えられる。さらに最近のメディアアートでは動作認識を利用したゲーム性のある作品が数多く発表されている。このようなジェスチャ認識を用いたインタフェースには提案手法を用いることによって、ユーザと作品のインタラクティブ性を高めることができると考えられる。

## 6. まとめ

本研究では、加速度センサを用いたジェスチャ認識における早期認識手法を提案した。早期かつ適切なタイミングで認識するために、入力データと教師データの距離を逐次的に計算し、最も近い距離と2番目に近い距離を相対的に判断することで確度の高い認識結果を得ることができる。評価から提案手法を用いることで、27種類のジェスチャ動作のほぼすべてを認識精度を低下させることなく、早期に認識できることを確かめた。

現段階では、確度の観点のみからジェスチャの認識結果のタイミングを決定しているため、高い確度が最後まで得

られないと認識結果の出力が動作終了後になる場合がある。今後は、拮抗している候補からジェスチャの残り時間を予測し、結果の出力が遅れるのであれば、高い確度が得られなかった場合でも、その時点において最も確度が高いジェスチャを出力する手法を提案する予定である。また、今回は処理時間をPC上で計測したが、スマートフォンなどの非力なCPUでは処理時間がセンサのサンプリング周期を上回り、逐次的な計算ができなくなる可能性があるため、動的に逐次処理の頻度を変更する手法を検討する予定である。

**謝辞** 本研究の一部は、科学技術振興機構さきがけおよびMicrosoft Research Asia Windows Phone Academic Programの支援によるものである。ここに記して謝意を表す。

## 参考文献

- [1] Tomibayashi, Y., Takegawa, Y., Terada, T. and Tsukamoto, M.: Wearable DJ System: a New Motion-Controlled DJ System. *Proc. The International Conference on Advance in Computer Entertainment Technology (ACE 2009)*, pp. 132–139 (2009).
- [2] Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J. and Vasudevan, V.: uWave: Accelerometer-based Personalized Gesture Recognition and its Applications, *Proc. IEEE International Conference on Pervasive Computing and Communication (PerCom 2009)*, pp. 1–9 (2009).
- [3] Agrawal, S., Constandache, I., Gaonkar, S., Choudhury, R., Caves, K. and DeRuyter, F.: Using mobile phones to write in air, *Proc. The International Conference on Mobile Systems, Applications, and Services (Mobisys 2011)*, pp. 15–28 (2011).
- [4] Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M. and Hashizume, H.: Toss-it: Intuitive Information Transfer Techniques for Mobile Devices, *Proc. The International Conference on Human Factors In Computing Systems (CHI 2005)*, pp. 1881–1884 (2005).
- [5] Kanke, H., Takegawa, Y., Terada, T. and Tsukamoto, M.: Airstic Drum: a Drumstick for Integration of Real and Virtual Drums, *Proc. The International Conference on Advance in Computer Entertainment Technology (ACE 2012)*, pp. 57–69 (2012).
- [6] Fujimoto, M., Fujita, N., Takegawa, Y., Terada, T. and Tsukamoto, M.: A Motion Recognition Method for a Wearable Dancing Musical Instrument, *Proc. The International Symposium on Wearable Computers (ISWC 2009)*, pp. 11–18 (2009).
- [7] Murao, K., Terada, T., Yano, A. and Matsukura, R.: Evaluation Study on Sensor Placement and Gesture Selection for Mobile Devices, *Proc. The International Conference on Mobile and Ubiquitous Multimedia (MUM 2012)*, No. 7 (2012).
- [8] Myers, C.S. and Rabiner, L.R.: A Comparative Study of Several Dynamic Time Warping Algorithms for Connected Word Recognition, *The Bell System Technical Journal*, Vol. 60, pp. 1389–1409 (1981).
- [9] Wireless Technologies, Inc.: available from (<http://www.wireless-t.jp/>)