Vol.2018-UBI-58 No.10
2018/5/19

ジェスチャ・行動認識のための加速度信号アップサンプリング手法に関する検討

吉村 直也[1,a)]　前川 卓也[1,b)]　天方 大地[1,c)]　原 隆浩[1,d)]

概要：スマートウォッチの普及に伴い，デバイスに搭載されたセンサを用いたジェスチャ認識の研究が盛んに行われている．近年の研究では，通常のスマートウォッチより高いサンプリングレートで計測した加速度信号を用いることで，手首に装着したセンサでも手先や指先の動きを伴う細かなジェスチャが認識できることが報告されている．しかし市販のデバイスではアプリケーションが取得できる加速度信号のサンプリングレートには制限があり，高いサンプリングレートの加速度信号を使用することは難しい．

本研究ではニューラルネットワークを用いて加速度信号のアップサンプリングを行い，擬似的な高いサンプリングレートの信号を生成する「加速度信号超解像技術」を提案する．加速度信号は線形補間など従来手法でも高い精度で補間ができる滑らかな変化と，従来手法では再現が難しい衝撃や振動 などによる突発的な変化が存在する．提案手法では両方の変化のパターンを高い精度で補間するために，線形補間をニューラルネットワークで補正する手法を用いた．またアップサンプリングした信号にジェスチャ・行動認識を適用し，提案手法の有効性を確認した．

1. はじめに

近年のスマートフォンやスマートウォッチは多種多様なセンサを搭載しており，これらのセンサを用いてジェスチャ認識や行動認識が行われている．手首に装着したセンサでジェスチャ認識や行動認識を行う研究はユビキタスコンピューティングの分野で盛んに研究されている．加速度信号を使用する場合，多くのジェスチャは 100 Hz 未満のサンプリングレートの信号で十分に認識ができる．また多くのスマートウォッチで取得することができる加速度信号のサンプリングレートもおよそ 100 Hz に制限されている[*1]．しかし Laput ら [8] はスマートウォッチのデバイスドライバを改造することで，サンプリングレートの高い加速度信号を用いてジェスチャ認識などを行なった．「tap」や「flick」などの動きが小さなジェスチャはサンプリングレートの低い信号では認識が難しいが，サンプリングレートの高い信号を使用すると認識ができるようになることを報告した．

本研究ではスマートウォッチなどで利用できるサンプリングレートの低い加速度信号を，ニューラルネットワークを用いたアップサンプリングを行うことで擬似的な高いサンプリングレートの信号を生成する「加速度信号超解像手法 (AccSR)」を提案する．多くのスマートウォッチでは加速度信号のサンプリングレートがおよそ 100 Hz に制限されていることを踏まえ，提案手法では 2 倍の 200 Hz の信号を生成することを試みる．アップサンプリングされた信号はジェスチャ認識・行動認識・ゲームの入力など様々な応用が存在する．本研究では提案手法の有効性を検証するために，アップサンプリングされた信号を用いてジェスチャ認識と行動認識を行う．

加速度センサは腕を動かすなどのセンサの空間的「移動」に加えて，センサをタップした際などに観測される「衝撃・振動」の 2 種類の現象を x，y，x の 3 軸で捉えることができる．「移動」の場合では加速度の値はゆっくり変化するが，「衝撃・振動」では非常に短時間で変化する．このように加速度信号のアップサンプリングを行うためには継続時間が異なる 2 つ現象を同時に考慮すること必要がある．従来手法の線形補間などでは，比較的ゆっくり変化する「移動」に対して高い精度で補間することができるが，「衝撃・振動」といった突発的に発生する現象を再現できない．また加速度センサは 3 軸で同じ現象を捉えている．したがってある軸を補間する際に他の 2 軸を考慮することが有効で

[1] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[a)] yoshimura.naoya@ist.osaka-u.ac.jp
[b)] maekawa@ist.osaka-u.ac.jp
[c)] amagata.daichi@ist.osaka-u.ac.jp
[d)] hara@ist.osaka-u.ac.jp
[*1] https://source.android.com/compatibility/8.1/android-8.1-cdd

あると考えられるが，従来手法では 3 軸の関係性を考慮することが難しい．提案手法では線形補間をニューラルネットワークで補正するという手法を用いることで，3 軸の関係性を考慮した上で 2 つの現象を同時に再現した．

本研究の技術的貢献は以下の通りである．

(1) 加速度信号のサンプリングレートを 2 倍にするアップサンプリング手法「加速度信号超解像技術 (AccSR)」を提案する．筆者が知る限り，超解像技術を加速度信号に対して適用した初めての研究である．

(2) 提案手法で生成した加速度信号をジェスチャ認識と行動認識に適用し，その有効性を確認した．

2. 関連研究

2.1 超解像技術

超解像技術は画像分野を中心に以前から研究されており，低解像度の信号を高解像度の信号に変換する技術である．近年 CNN ベースのニューラルネットワークを用いた SRCNN[3] が提案され，性能が飛躍的に向上した．また画像の超解像技術を音声信号に応用して，入力音声をより明瞭な音声に再構成する audio super-resolution[7] も研究されている．画像と音声のいずれの超解像技術においても，処理後の信号は人間が知覚することを想定しており，信号全体を再構成するため入力信号が出力信号に保存されない．一方，本研究ではアップンサンプリングした信号を多種のアプリケーションに使用することを想定しており，抽出された特徴ではなく，より真値に近い時系列信号を生成することが目的である．したがって提案手法では先行研究のニューラルネットワークの構成などを参考にし，信号全体の再構成ではなく補間によって 100 Hz の信号を 200 Hz に変換した．

2.2 ジェスチャ認識

ジェスチャ認識はより自然で直感的なインタフェースを実現する技術として盛んに研究されてきた [11]．ジェスチャ認識はウェアラブルデバイスの操作やゲームの入力などに使用されている．多くの先行研究では，腕で空中に数字を書くなどの大きな腕の動きを伴うジェスチャの認識に注目が置かれており，これらのジェスチャは低いサンプリングレートの信号でも十分に認識ができる．一方で Laput ら [8] は「tap」や「flick」など指や手首を中心とした小さな動きのジェスチャの認識を試みた．デバイスのドライバを改造し，4000 Hz という非常に高いサンプリングレートの信号を使用することで，これらの動きの小さななジェスチャの認識が可能なことを報告した．本研究ではデバイスの改造ではなく仮想的なサンプリングレートの高い信号を生成することで，動きの小さなジェスチャの認識を試みた．

2.3 行動認識

行動認識は高齢者の見守りやライフログ生成などのために盛んに研究されている．スマートウォッチをはじめとするウェアラブルデバイスなどを用いて体の動きを捉える手法が，多くの行動認識の先行研究において用いられている [1], [2], [10]．一方で使用しているオブジェクトがそのユーザの行動を反映していることに注目し，ユーザが使用しているオブジェクトを認識することで行動認識を行う手法も研究されている [9]．本研究では電化製品に搭載されたモータが動作する際に発する微細な振動を捉え，ユーザが使用しているオブジェクトを認識することで行動認識を行なった．モータが発する振動に含まれる高周波成分を提案手法で復元することにより認識を試みた．

3. 加速度信号超解像手法

超解像技術やニューラルネットワークの先行研究を参考に，加速度信号の特性を考慮したアップサンプリングの手法を提案する．本研究では手首に装着したセンサで取得した加速度信号を 100 Hz から 200 Hz にアップサンプリングする．

3.1 アップサンプリング

本研究では 2.1 節で述べた通り，より真の 200 Hz 信号に近い信号を生成するため補間によるアップサンプリングを行う．

200 Hz と 100 Hz の加速度信号 $\boldsymbol{A}^H, \boldsymbol{A}^L$ は，200 Hz のサンプリング間隔を $\Delta = 0.05$[ms]，時刻 $n\Delta$ のにおける加速度 3 軸の値を $\boldsymbol{a}_{n\Delta}$ とすると，

$$\boldsymbol{A}^H = [\boldsymbol{a}_{0\Delta}, \boldsymbol{a}_{1\Delta}, \boldsymbol{a}_{2\Delta}, \ldots, \boldsymbol{a}_{n\Delta}, \ldots]$$
$$\boldsymbol{A}^L = [\boldsymbol{a}_{0\Delta}, \boldsymbol{a}_{2\Delta}, \ldots, \boldsymbol{a}_{2n\Delta}, \boldsymbol{a}_{(2n+2)\Delta}, \ldots]$$

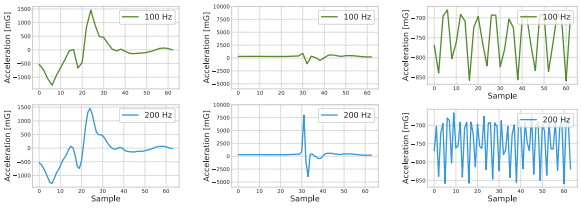
補間による 100 Hz から 200 Hz の 2 倍のアップサンプリングは，$\boldsymbol{A}^L$ における時刻 $2n\Delta$ と $(2n+2)\Delta$ の中間の時刻 $(2n+1)\Delta$ における加速度 3 軸の補間値を生成して $\boldsymbol{A}^L$ に挿入する処理である．ここで生成された時刻 $n+\Delta$ における補間値を $\hat{\boldsymbol{a}}_{(2n+1)\Delta}$ とすると，アップサンプリングされた信号 $\boldsymbol{A}^{SR}$ は以下のようになる．

$$\boldsymbol{A}^{SR} = [\boldsymbol{a}_0, \hat{\boldsymbol{a}}_1, \ldots, \boldsymbol{a}_{(2n-2)\Delta}, \hat{\boldsymbol{a}}_{(2n-1)\Delta}, \boldsymbol{a}_{2n\Delta}, \ldots]$$

単純な補間手法として線形補間がある．時刻 $(2n+1)\Delta$ における補間値 $\hat{\boldsymbol{a}}_{(2n+1)\Delta}$ を生成する場合，補間値は直前直後の時刻の加速度の値の平均として以下の式で計算される．

$$\hat{\boldsymbol{a}}^{linear}_{(2n+1)\Delta} = \frac{1}{2}\left(\boldsymbol{a}_{2n\Delta} + \boldsymbol{a}_{(2n+2)\Delta}\right) \quad (1)$$

本研究は動きの小さなジェスチャを認識することを目的としており，衝撃などの高周波成分を再現することが重要である．しかし線形補間は高速であるが，前後の値を平均値を用いて補間するため，出力信号 $\boldsymbol{A}^{SR}$ 全体が本来の信号



(1) 腕の動きによる滑らかな変化　(2-1) 衝撃による突発的な変化　(2-2) 電子機器などによる振動

図 1　加速度信号の変化の傾向．上段が 100 Hz，下段が 200 Hz．

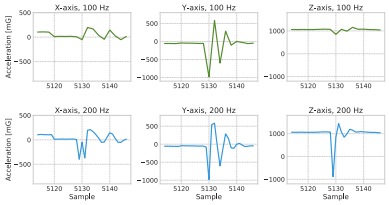


図 2　加速度信号の 3 軸の相関関係．ある軸で捉えられていない情報を他の軸が捉えている可能性がある．

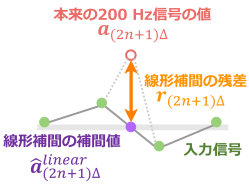


図 3　線形補間の残差

$\boldsymbol{A}^H$ よりも滑らかになる．したがって線形補間では本研究の目的に対して十分正確なアップサンプリングができない．

3.2　加速度信号の特徴

図 1 に示すように，加速度信号の波形は (1)「滑らかな変化」と (2)「衝撃や振動」の 2 種類に大別される．加速度センサを手首に装着している場合，腕を振るなどセンサの「移動」を観測した際に加速度の値は滑らかな変化をする．また「衝撃や振動」を示す波形は，手を叩いたり腕をぶつける，あるいは電化製品を使用した際に現れる．線形補間は (1)「滑らかな変化」を小さな誤差で補間できるが，(2)「衝撃や振動」は他と波形が大きく異なり，さらに突発的に発生するので (1) 滑らかな波形と同じアルゴリズムだけでは再現できない．

また加速度信号は同じ現象を 3 軸で捉えているので，x，y，z の 3 軸には相関関係がある．図 2 に衝撃を捉えた加速度信号の波形を示す．200 Hz の信号では全ての軸で衝撃を捉えている．しかし 100 Hz では衝撃を y 軸でははっきり捉えているが，z 軸において衝撃をはっきりと捉えることができていない．このようにある軸で捉えられていない情報を他の軸が捉えている場合が多く存在するが，線形補間やスプライン補間では補間したい軸以外の波形を考慮することができない．正確な補間を行うためには 3 軸の関係性を考慮することが重要であると考えられる．

3.3　提案手法

このように 2 種類の現象を同時に高い精度でアップサンプリングするには，補間したい点の周辺の時間的な変化や 3 軸の変化などの複合的な情報を考慮する必要がある．提

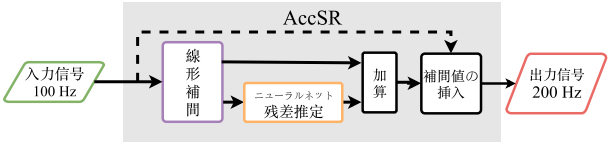


図 4　提案手法の概要．提案手法ではまず入力信号に対し線形補間と，ニューラルネットワークによる線形補間の残差推定を行う．最後に線形補間の補間値と推定した残差の和を提案手法の補間値として，入力信号に挿入する．

案手法の処理フローを図 4 に示す．提案手法では線形補間をニューラルネットワークを用いて補正することで，線形補間では捉えられない (2) 衝撃や振動を再現する．線形補間を補正するためのニューラルネットワークでは，図 3 に示す線形補間からの誤差である残差を推定する．提案手法では補間値を各軸ごとに 1 点ずつ順番に計算し，生成された補間値を $\boldsymbol{A}^L$ に挿入していくことでアップサンプリングされた信号 $\boldsymbol{A}^{SR}$ を生成する．

提案手法の補間値 $\hat{\boldsymbol{a}}^{AccSR}_{(2n+1)\Delta}$ の計算方法を説明する．補間値 1 点の計算には補間する点の前後 k 点から成る $(2k \times 3)$ 次元のベクトル $\boldsymbol{x}_{(2n+1)\Delta}$ を用いる．提案手法では $2k = 32$ を用いた．提案手法ではまず式 1 によって線形補間の補間値 $\hat{\boldsymbol{a}}^{linear}_{(2n+1)\Delta}$ を計算する．次にニューラルネットワークの入力として式 2 のように，$\boldsymbol{x}_{(2n+1)\Delta}$ の各要素から線形補間の補間値を減算した

$$\boldsymbol{x}'_{(2n+1)\Delta} = \boldsymbol{x}_{(2n+1)\Delta} - \hat{\boldsymbol{a}}^{linear}_{(2n+1)\Delta} \tag{2}$$

を計算する．加速度信号は重力などのバイアスを常に受けている．このバイアスはデバイスの姿勢推定などには有用であるが，衝撃や振動を再現するためには必要でない．よってこの前処理によって加速度信号がうけるバイアスの影響を除去している．ニューラルネットワークでは $\boldsymbol{x}'_{(2n+1)\Delta}$ から残差，

$$\boldsymbol{r}_{(2n+1)\Delta} = \boldsymbol{a}_{(2n+1)\Delta} - \hat{\boldsymbol{a}}^{linear}_{(2n+1)\Delta} \tag{3}$$

を推定する．ニューラルネットワークによって求めた残差の推定値を $\hat{\boldsymbol{r}}_{(2n+1)\Delta}$ とする．

最後に線形補間によって求めた補間値とニューラルネットワークの出力の推定残差を足し合わせ補間値 1 点の値，

$$\hat{\boldsymbol{a}}^{AccSR}_{(2n+1)\Delta} = \hat{\boldsymbol{a}}^{linear}_{(2n+1)\Delta} + \hat{\boldsymbol{r}}_{(2n+1)\Delta} \tag{4}$$

を求める．提案手法の補間値は最終的には式 4 となる．計算した補間値を 100 Hz の入力信号に挿入することでアップサンプリングされた加速度信号を生成する．

3.4　ネットワーク構成

提案手法の線形補間の補間値からの残差を推定するニューラルネットワークの構成を図 5 に示す．ネットワークへの入力は式 2 の $(2k \times 3)$ 次元ベクトルで，出力は式 3 の残差である．ネットワークは特徴抽出のための分岐した畳み込

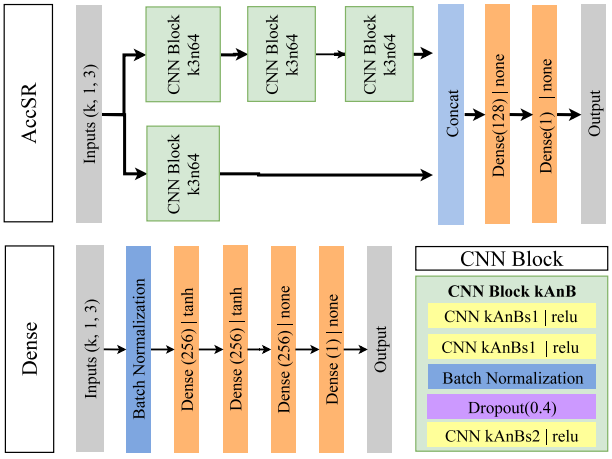


図 5 ニューラルネットワークの構成

み層からなる部分と出力層の全結合層 (Dense) 2 層で構成されている．

特徴抽出を行う分岐構造部分はそれぞれ畳み込み層 3 層から成るブロックを 3 段と 1 段を積み重ねた構成になっている．それぞれのブロックは畳み込み層 3 層から構成されており，各ブロックの上層 2 層はストライドを 1 に設定し，時間方向の広がりを保ったまま特徴抽出を行う．一方 3 層目はストライドを 2 に設定することで時間方向に圧縮をかけ，より抽象度の高い特徴を抽出する．またブロック 1 段からなる分岐部分は 7×1 の大きい畳み込みフィルタを用いることで入力信号の大域的な特徴を捉える．もう一方の分岐部分はより小さな 3×1 の畳み込みフィルタを使用して局所的な特徴を捉えるとともに，ブロックを 3 段重ねる深い構造をとることでより複雑な特徴を抽出する．各分岐部分の 1 段目にはフィルタ数として 32 を用いた．各ブロックの最後に行う時間方向の圧縮で表現の自由度が減るので，次のブロックでは前のブロックの 2 倍のフィルタ数を用いることで表現できる特徴の自由度を回復している．したがって右分岐の 2 段目，3 段目のフィルタ数はそれぞれ 64，128 である．このように右分岐部分では圧縮と自由度の回復を繰り返すことで密度の高い高次な特徴を抽出する．

分岐部分によって計算された特徴は結合され全結合層 (Dense layer) に渡され，最終的な出力が計算される．この全結合層には活性化関数を使用していない．

3.5 ニューラルネットワークの学習

ニューラルネットワークの学習は，用途 (ジェスチャ認識，オブジェクト認識) と加速度の軸 (x, y, z) 毎に行い，それぞれ個別にモデルを作成した．

ジェスチャ認識用のモデルでは，まず日常生活中に計測したデータを用いて事前学習を行なった．この追加データははには食事やタブレット端末でゲームをプレイしているデータなどが含まれている．事前学習を行なった後，ジェスチャのラベル付きデータと事前学習で利用したデータの

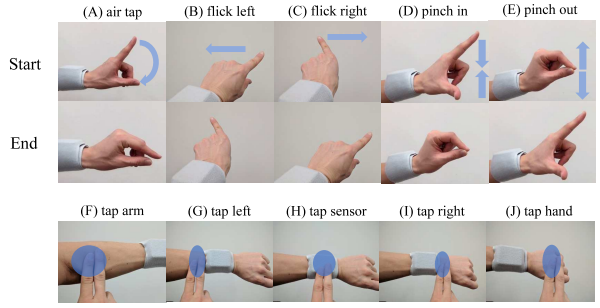


図 6 実験で使用したジェスチャ

一部を用いてファインチューニングを行なった．事前学習は 30 epoch，ファインチューニングは 10 epoch 行った．日常生活中に計測した追加データは合計で約 6 時間半，ラベル付きのデータは約 1 時間半である．

オブジェクト認識用のモデルもジェスチャ認識用のモデルと同様に事前学習とファインチューニングを行った．事前学習には食事中のデータと掃除中のデータを用いた．ファインチューニングにはオブジェクト使用中のデータと事前学習用データの一部を使用して学習した．事前学習は 30 epoch, ファインチューニングは 5 epoch 行った．日常生活中に計測した追加データは約 2 時間半，オブジェクト使用中のラベル付きデータは 1 時間程度の長さである．

ネットワークの最適化には平均絶対誤差 (MAE) を損失関数として使用した．Adam Optimizer[6] を使用してバッチサイズ 256 のミニバッチ学習を行い，過学習の防止と学習の効率的な進行のため Dropout と Batch Normalization を用いた．Dropout と Batch Normalization の適用方法は提案手法と同じく複数の畳み込み層をブロックとして組織化した構造を持つ WideResNet[13] や Hi [5] らの研究を参考にして，図 5 に示すように 2 層目と 3 層目の畳み込み層の中間に配置した．

4. 評価実験

4.1 データセット

4.1.1 ジェスチャと行動認識のセット

本研究で使用したジェスチャ 10 種類を表 7 に示す．本研究で使用するジェスチャは [8], [12] を参考に，スマートフォンや VR デバイスで使用されている操作から選択した．ジェスチャの実施方法は図 6 に示す．実験では 5 人の被験者に 10 種類のジェスチャを 1 セッションにつきランダムな順番に 2 回ずつ，合計 10 セッション行なってもらった．

電化製品の利用履歴に基づく行動認識のために用いた全 9 種類のオブジェクトを表 8 に示す．使用時はセンサを装着している左手でオブジェクトを持ち，電源を入れいている間のみ使用中と判断した．それぞれのオブジェクトには利用時に固有の動作が存在するが，本研究ではモータが発する微細な振動を捉えて認識可能かを調べるため使用時は

図 7 行動認識に使用したジェスチャ

	Gesture (サンプル数)		Gesture (サンプル数)
A	air tap (89)	F	tap arm (98)
B	flick left (99)	G	tap left (97)
C	flick right (97)	H	tap sensor (98)
D	pinch in (96)	I	tap right (98)
E	pinch out (99)	J	tap hand (98)

図 8 行動認識に使用したオブジェクト

	Object (サンプル数)		Object (サンプル数)
A	blender (40)	F	handy vacuum (40)
B	drill (40)	G	pepper mill (39)
C	screwdriver (40)	H	shaver (40)
D	fun (38)	I	toothbrush (40)
E	handy blender (40)		

オブジェクトを持っている腕を静止させた．実験では 5 人の被験者に 1 セッションにつきランダムな順番で 1 回ずつ，合計 5 セッション，オブジェクトを利用してもらった．

表 7 と表 8 に本実験で使用したサンプル数を示した．被験者が指示されたジェスチャまたはオブジェクトとは異なるものを行ったなどの理由で，サンプル数に偏りが生じている．

4.1.2 データ収集

前節で説明したジェスチャおよびオブジェクトの利用を 5 人の被験者に行ってもらい，左手にバンドで装着した 3 軸加速度センサ (ATR TSND151[*2]) で実験に使用する加速度信号を記録した．また加速度信号にラベル付けを行うため同時にビデオも記録した．本研究では同じ位置で計測された 2 種類の異なるサンプリングレートの信号が必要であるが，同じ場所に 2 つのセンサを装着することは不可能である．このため 1000 Hz のサンプリングレートで加速度信号を記録し，1000 Hz の信号を間引きすることで 200 Hz と 100 Hz の信号を生成した．1000 Hz から 200 Hz のダウンサンプリングを行うと，間引きの開始点を変えることで 200 Hz の信号が 5 系列生成される．ニューラルネットワークの学習には 5 系列全てを使用し，それ以外は 1 系列のみを使用して学習・評価を行った．

記録した加速度データに対して，同時に記録したビデオを確認してラベル付けを行った．ジェスチャは「tap」など非常に短く終了点がわからないものが多いため，ジェスチャの開始点からおよそ 0.3 秒に対してラベルを付与した．行動認識用のオブジェクト利用は，オブジェクトの電源が入っている時にラベルを付与した．

4.2 補間精度

4.2.1 評価方法

提案手法の補間精度を leave-one-participant-out 交差検定によって検証した．3.5 節に示した方法で日常生活中に記録したデータを事前学習したニューラルネットワークを，被験者 1 人分のデータを除いた残りの被験者のデータを用いてファインチューニングを行い，除外した信号に対しアップサンプリングを行い評価を行った．また学習はジェスチャ認識用とオブジェクト認識用に分けて行い，評価にはラベルが付与されている部分のみを使用した．

補間した加速度信号の評価には，平均絶対誤差 (MAE [0.1mG])，平均 2 乗誤差 (MSE $[(0.1mG)^2]$)，signal-to-noise ratio (SNR [dB])，log-spectrum distance (LSD [dB]) を使用した．これらの評価指標は各軸ごとに各ジェスチャ・オブジェクト使用に対して抽出したウィンドウに対して計算した．MAE，MSE，SNR は以下の式によって計算される．ただし，真の 200 Hz の信号系列を $\boldsymbol{x}$，アップサンプリングされた信号系列を $\hat{\boldsymbol{x}}$，時刻 $n\Delta$ における信号 $\boldsymbol{x}$ の値を $x_{n\Delta}$ とする．

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=0}^{N}|\hat{x}_{n\Delta} - x_{n\Delta}| \tag{5}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=0}^{N}(\hat{x}_{n\Delta} - x_{n\Delta})^2 \tag{6}$$

$$\mathrm{SNR}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = 10\log\frac{||\boldsymbol{x}||^2}{||\hat{\boldsymbol{x}} - \boldsymbol{x}||_2^2} \tag{7}$$

LSD は周波数領域において各周波数成分の復元精度を測る指標であり [4]，以下の式で計算される．$\boldsymbol{X}$，$\hat{\boldsymbol{X}}$ はぞれぞれ $\boldsymbol{x}$，$\hat{\boldsymbol{x}}$ の対数パワースペクトルを，l，k はそれぞれフレームと周波数成分のインデックスを表す．

$$\mathrm{LSD} = \frac{1}{L}\sum_{l=0}^{L}\sqrt{\frac{1}{K}\sum_{k=0}^{K}\left(\hat{\boldsymbol{X}}(l,k) - \boldsymbol{X}(l,k)\right)^2} \tag{8}$$

提案手法の有効性を確認するため，以下の補間手法と比較を行った．

- **AccSR**: 提案手法
- **AccSR (Single)**: 提案手法の入力を 3 軸から残差を推定する 1 軸のみとした．
- **Dense**: 図 5 に示す全結合層のみからなるニューラルネットワークを用いて直接補間値の推定を行った．
- **線形補間 (Linear)**: 式 1 に示す線形補間で補間値を推定した．
- **スプライン補間 (Spline)**: 3 次元スプライン関数を使用して補間を行った．入力は提案手法と同じである．

4.2.2 結果

アップサンプリングした信号全体に対する評価結果を表 9 と表 10 に示す．提案手法が線形補間やスプライン補間に比べて大きく補間誤差を削減した．線形補間に対する提案手法の MSE は．ジェスチャ認識用のデータでは半分程度，オブジェクト認識用のデータでは 1/7 程度になった．また Dense と比較して提案手法はオブジェクト認識用データで MSE が 1/3 程度になっており，提案の残差推定を行

[*2] http://www.atr-p.com/products/TSND121.html

うニューラルネットワークとその構成という提案手法の優位性が確認された．また LSD についても提案手法が線形補間やスプライン補間に対して誤差を小さくしており，周波数領域においても精度の高いアップサンプリングが行えたと言える．SNR もわずかであるが提案手法が線形補間やスプライン補間よりも大きくなっており，信号中のノイズの比率が小さくなったことが分かる．オブジェクト認識用データと比較してジェスチャ認識用データの誤差の削減幅が小さくなっている．これは振動を捉えている部分のみを使用しているオブジェクト認識用のデータに比べて，ジェスチャ認識用データにはジェスチャの前後の静止状態が含まれており，静止状態では加速度の値の変化が小さい．したがって線形補間でも十分に高い精度で補間ができるため，全体として提案手法との差が小さくなっていると考えられる．また，オブジェクト認識用データにおいて比較手法の AccSR (Single) の方が提案手法の AccSR より MAE，MSE が小さくなった．オブジェクト使用中に観測される振動は定常的なものであり，1 軸の情報のみで十分に振動を捉えて補間ができたと考えられる．ジェスチャ認識では提案手法の AccSR の方が，MAE・MSE において精度の高い補間ができていた．

アップサンプリングを行った「tap hand」と「toothbrush」の加速度信号を図 11[1-2] に示す．「tap sensor」では 100 Hz では捉えられていない突発的な変化を提案手法では再現できた．また「toothbrush」では，100 Hz では捉えられていない激しい振動を提案手法では再現できた．図 11[3-4] に「pepper mill，tap hand」を AccSR，Dense，線形補間によってアップサンプリングした x 軸の加速度信号を示した．「pepper mill」の加速度信号は滑らかに変化しているが，Dense が生成した信号には微細なノイズが発生している．Dense では補間値を直接推定しているため提案手法に比べてニューラルネットワークが行う処理が多く，「滑らかな変化」と「振動」を正しく区別するための特徴を正しく学習することが難しかったと考えられる．さらに「振動」の方が予測が難しく，誤差を大きくしやすいため，「振動」に過剰に反応するような学習が行われ，このような微細な振動が生成されたと考えられる．同様の現象が「tap hand」などジェスチャ認識用のデータでも発生した．

4.3 ジェスチャ認識性能

4.3.1 評価方法

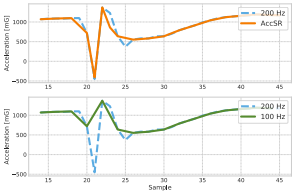
提案手法でアップサンプリングした加速度信号に対しジェスチャ認識を行い，その認識精度の評価を行った．ジェスチャ認識では認識対象となる加速度信号から，各軸の 128 点を含む 128×3 次元のウィンドウ ($\boldsymbol{W}_{t'}$) を作成し，各軸に対して MFCC およびウィンドウ内の平均，最大，最小，最大 - 最小，標準偏差を計算した．その後 leave-one-session-out 交差検定によって，Random Forest を使用して学習と認識を行った．Random Forest の学習には提案手法によってアップサンプリングした信号 $\boldsymbol{A}^{SR}$ を用いた．

図 9 補間性能: Gesture

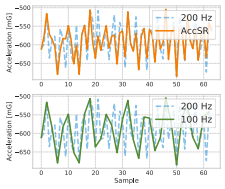
Method	MAE	MSE	SNR	LSD
AccSR (提案手法)	207.7	4918361.5	29.5	7.07
AccSR (Single)	213.0	5241043.8	29.6	7.01
Dense	327.2	5653697.9	23.6	7.60
Linear	267.8	9483778.8	28.4	11.87
Spline	287.3	11032662.0	29.2	20.23

図 10 補間性能: Object

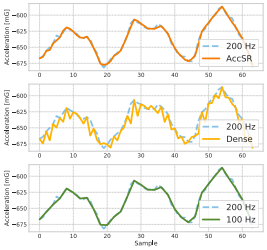
Method	MAE	MSE	SNR	LSD
AccSR (提案手法)	64.5	23430.0	38.2	8.47
AccSR (Single)	64.3	22771.8	37.5	8.49
Dense	117.9	70727.6	32.3	10.02
Linear	158.0	158440.4	33.9	12.40
Spline	186.4	225130.6	33.9	18.24



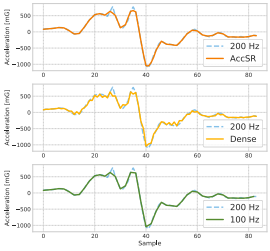
[1] tap hand (z 軸)



[2] toothbrush (x 軸)



[3] pepper mill (z 軸)



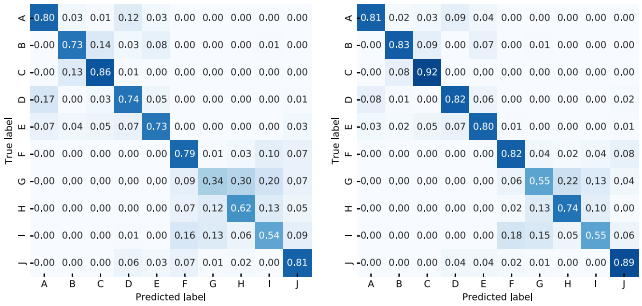
[4] flick left (z 軸)

図 11 アップサンプリングされた加速度信号

4.3.2 結果

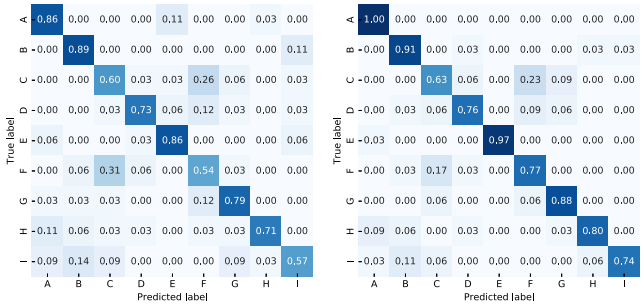
ジェスチャ認識の結果とその混合行列を表 1 と図 12 に示す．混合行列は行方向の値の和が 1 になるように正規化した．100 Hz の信号で認識を行った場合に比べて，提案手法の方が 10 種類のジェスチャ平均の F 値で平均 0.08 以上認識精度が向上した．また 200 Hz で認識した場合に対しては差が 0.03 と 100 Hz で認識した場合に比べて小さな差になった．提案手法がジェスチャ認識において有効であると言える．また「(E) pinch out」を除く全てのジェスチャで AccSR (Single) よりも提案手法の方が高い認識精度となった．したがって加速度のアップサンプリングにおいて 3 軸の関係も考慮に入れることは重要であると言える．

個別のジェスチャでは特に「(G) tap left」と「(H) tap sensor」が F 値で 0.10 以上の大きな認識精度の向上が見られた．混合行列から似た種類のジェスチャ (ex. tap に関



[1] 100 Hz [2] AccSR

図 12 ジェスチャ認識結果の混合行列



[1] 100 Hz [2] AccSR

図 13 オブジェクトベース行動認識結果の混合行列

するジェスチャ同士) の誤分類が減少していることがわかる．これは提案手法により衝撃の情報が再現されたため分類できるようになったと考えられる．

4.4 行動認識性能

4.4.1 評価方法

アップサンプリングした加速度信号に対してオブジェクトベースの行動認識を行い，認識精度を評価する．ジェスチャ認識と同様に各ラベルに対応する信号からウインドウ $\boldsymbol{W}_{t'}$ を作り特徴抽出を行った後，Random Forest で認識を行った．評価は leave-one-participant-out 交差検定を用いた．アップサンプリングした加速度信号は，leave-one-participant-out 交差検定のために学習したモデルで作成した．

4.4.2 結果

オブジェクトベースの行動認識を行った結果の表と混合行列を表 2 と図 13 に示す．提案手法は 100 Hz の信号で認識を行った場合より 9 種類のオブジェクトの平均の F 値で 0.10 以上認識精度が向上した．200 Hz の認識結果と提案手法では 0.003 と非常に小さな差となった．「(A) blender」など 5 種類のオブジェクトで認識精度が 200 Hz の信号を用いた場合より提案手法の方が高い値となった．これはアップサンプリングがノイズキャンセリングの働きをしたためと考えられる．

個別のオブジェクトでは「(I) toothbrush」が F 値で 0.20 以上と認識精度が特に大きく向上した．混合行列より 100 Hz の信号を用いた場合「(I) toothbrush」を「(A) blender, (B) drill, (C) screwdriver, (G) pepper mill」と誤認識されていたが，提案手法においてその誤認識の割合が減少していることがわかる．

4.5 計算時間

4.5.1 評価方法

ニューラルネットワークの補間値 1 点を計算するためにかかる計算時間を計測した．計測にはジェスチャの補間精度を評価したものと同じデータを用い，ニューラルネットワークによる複数点の残差計算の時間を測定し，それをサンプル数で割ることで 1 点あたりの計算時間を求めた．計算は GeForce GTX TITAN X Graphics Card[*3]を用いて行った．

4.5.2 結果

各軸における 1 点あたりの計算時間は平均 0.25 [ms] であり，1 時刻分の 3 軸全ての残差の計算にはおよそ 0.76 [ms] かかる．この残差の計算時間は 100Hz におけるサンプリング間隔の 10.0 [ms] より十分小さいため，ニューラルネットの計算で補間に遅延が生じる可能性は少なくリアルタイム性が求められるアプリケーションにも十分利用可能であると言える．また今回の計測には GPU を使用したが，現在多くのスマートフォンには GPU が搭載されていない．しかし iPhone X のプロセッサに搭載された「ニューラルエンジン」のように，高速かつ低消費電力で機械学習の計算を処理する専用の回路が開発されている．近い将来にスマートフォン上でも GPU と同等の速度でニューラルネットワークの処理を実行できると考えられる．

5. 結論

本研究ではニューラルネットワークを用いた加速度信号のアップサンプリング手法「加速度信号超解像技術 (AccSR)」を提案した．提案手法では加速度信号が捉える現象の特性や 3 軸の関係性などを考慮し，線形補間をニューラルネットワークで補正するという手法で高い補間精度を実現した．また提案手法でアップサンプリングされた信号をジェスチャ認識と行動認識に適用し，認識精度が向上することを確認した．

今後の研究課題として，提案手法はニューラルネットワークを常時使用するためモバイル機器で使用する場合には消費電力が問題になる．現在消費電力を評価できていないのでまずこの評価を行うとともに，より低消費電力で動作できるアルゴリズムを検討していきたい．

[*3] http://www.nvidia.co.jp/object/geforce-gtx-titan-x-jp.html

表 1 アップサンプリングした加速度信号を使用したジェスチャ認識結果

		AccSR			AccSR (Single)	100 Hz	200 Hz
	Gesture	Precision	Recall	F1	F1	F1	F1
A	air tap	0.867	0.809	0.837	0.816	0.776	0.807
B	flick left	0.862	0.827	0.844	0.812	0.758	0.821
C	flick right	0.840	0.918	0.877	0.866	0.818	0.822
D	pinch in	0.806	0.823	0.814	0.796	0.724	0.860
E	pinch out	0.788	0.804	0.796	0.811	0.759	0.842
F	tap arm	0.734	0.816	0.773	0.752	0.720	0.827
G	tap left	0.609	0.546	0.576	0.488	0.420	0.659
H	tap sensor	0.723	0.745	0.734	0.656	0.613	0.843
I	tap right	0.667	0.551	0.603	0.589	0.546	0.641
J	tap hand	0.806	0.888	0.845	0.765	0.756	0.867
	平均	0.770	0.773	0.770	0.735	0.689	0.799

表 2 アップサンプリングした加速度信号を用いたオブジェクト認識の結果

		AccSR			AccSR (Single)	100 Hz	200 Hz
	Object	Precision	Recall	F1	F1	F1	F1
A	blender	0.875	1.000	0.933	0.796	0.800	0.877
B	drill	0.800	0.914	0.853	0.849	0.816	0.933
C	screwdriver	0.647	0.629	0.638	0.864	0.575	0.620
D	fun	0.833	0.758	0.794	0.606	0.774	0.900
E	handy blender	1.000	0.971	0.986	0.814	0.833	0.943
F	handy vacuum	0.675	0.771	0.720	0.943	0.528	0.658
G	pepper mill	0.833	0.882	0.857	0.600	0.783	0.800
H	shaver	0.903	0.800	0.848	0.857	0.806	0.896
I	toothbrush	0.963	0.743	0.839	0.831	0.635	0.871
	平均	0.837	0.830	0.830	0.796	0.728	0.833

謝辞

本研究の一部は JST CREST JPMJCR15E2, JSPS 科研費 JP16H06539，JP 17H04679 の助成を受けて行われたものです.

参考文献

[1] L. Bao and S. Intille, “Activity recognition from user-annotated acceleration data,” Int’l Conf. on Pervasive computing, pp.1–17, 2004.
[2] M. Berchtold, M. Budde, D. Gordon, H.R. Schmidtke, and M. Beigl, “Actiserv: Activity recognition service for mobile phones,” Int’l Symposium on Wearable Computers, pp.1–8, IEEE, 2010.
[3] C. Dong, C.C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.38, no.2, pp.295–307, 2016.
[4] A. Gray and J. Markel, “Distance measures for speech processing,” IEEE Trans. Acoustics, Speech, and Signal Processing, vol.24, no.5, pp.380–391, 1976.
[5] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” European Conf. on Computer Vision, pp.630–645, 2016.
[6] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
[7] V. Kuleshov, S.Z. Enam, and S. Ermon, “Audio Super Resolution using Neural Networks,” the 5th Int’l Conf. on Learning Representations, aug 2017.
[8] G. Laput, R. Xiao, and C. Harrison, “Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers,” the 29th Annual Symposium on User Interface Software and Technology, pp.321–333, 2016.
[9] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai, “Recognizing handheld electrical device usage with hand-worn coil of wire,” Pervasive 2012, pp.234-252, 2012.
[10] T. Maekawa and S. Watanabe, “Unsupervised activity recognition with user’s physical characteristics data,” Annual Int’l Symposium on Wearable Computers, pp.89–96, 2011.
[11] S.S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: a survey,” Artificial Intelligence Review, vol.43, no.1, pp.1–54, 2015.
[12] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, “Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum,” the 29th Annual Symposium on User Interface Software and Technology, pp.851–860, 2016.
[13] S. Zagoruyko and N. Komodakis, “Wide residual networks,” CoRR, vol.abs/1605.07146, 2016.