

加速度と音の振幅を用いて机へのタップを非接触判定するアプリの開発

片山 唯佳^{1,a)} 勝間 亮^{1,b)}

概要：現在、加速度センサとマイクが搭載されたスマートフォンやタブレット等のモバイル機器が広く普及してきている。本研究では、このモバイル機器を机の上に置いた状態で、人による机へのダブルタップを検知することを目的とする。加速度がある閾値を超えたかどうかでタップ判定し、タップ判定の後に待ち時間を設ける既存のタップ判定方法では、タップする力の強弱が変わると誤判定が起きやすい。そこで、本論文では加速度と音を併用し、振動の減衰率を用いることでタップの強弱に頑強かつリアルタイムにタップ判定を行うアプリを開発した。実験では、既存の手法と提案手法の両方でタップ判定を行い、シングルタップの精度では既存手法の72%に対して提案手法は98%を達成し、ダブルタップでも提案手法の方が精度よく判定できることを確認した。

1. はじめに

音楽と共に画面上に表示されるオブジェクト（ノーツ）がユーザのアクションを促し、ユーザが行った指定のアクションのタイミングの良さをスコア化し、より高スコアを目指す音楽ゲームがある。音楽ゲームの種類によって必要とするユーザのアクションは変わり、手を使って指定位置にタップする、ボタンを押す、足でパネルを踏むなどがある。いくつかの音楽ゲームではアクションを行ったタイミングが指示されたタイミングから前後に約0.1秒、約0.066秒、約0.033秒以内だったとき、この順に高得点になると言われている。基本的には1つのノーツで1つのアクションとその実行タイミングが指示され、1つの楽曲の中でいくつもノーツが出現するため、ユーザは何度も正しいアクションを正しいタイミングで行うことにチャレンジすることになる。

ユーザのアクションを要求するノーツの配置を譜面といい、様々な曲や譜面がある。初心者向けは低密度で単純な配置であり、熟練者向けになるにしたがって高密度で複雑な配置となっていく、高難度化する傾向がある。音楽ゲームは専用の大型機器を用いることが多く、ゲームセンターで遊ぶことを基本としているユーザは多い。高難度の譜面で良いスコアを狙うには相当数の練習が必要なため、ゲームセンター以外の場所ではプレイヤーはYoutube等で上級

者のプレイ動画や譜面の動画を流しながら、机などをタッチパネル、ボタン等に見立てて指で叩くことで、アクションを起こすタイミングについての練習をすることがある。しかし、練習の時に本当に正しいタイミングで叩いているかを判定するものが現在ない。

正しいタイミングで机を叩いているかを知るためには、机を叩いた時刻を特定する手法と、叩いたタイミングが正しいかどうか判定する手法が必要である。本稿では前者に焦点を当て、静かな環境でタブレットを使用しタップ時の入力波を測定することでユーザーが机を叩いた動作をリアルタイムに判定することを目標とし、まずは一定時間の間に2回タップする動作であるダブルタップをできるだけ正確に判定する。

加速度を使ったアクション認識手法は多く提案されている [1][2][3]。その中で本研究の対象であるタップ動作の検出について、手に装着した加速度センサから得られた加速度データを使用するものがある [4][5]。しかし、音楽ゲームの練習というシチュエーションでは手にセンサを装着することは動作の妨げになってしまうため避ける必要がある。また、加速度センサを直接タップした時の加速度を使って判定する方法がある [6]。しかし、音楽ゲームで行うタップは力強いいため、センサを直接タップすると傷ついてしまう可能性がある。そこで、直接センサをタップしない方式として机にタブレットを置き、机を叩いた時の加速度を測定する環境を想定する。また、既存の手法では、タップが行われたと判定された後、一定の待ち時間は次のタップ判定は行われぬ。しかし、実際の音楽ゲームのタップ間隔は

¹ 大阪府立大学
Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

^{a)} sbb01050@edu.osakafu-u.ac.jp

^{b)} katsuma@cs.osakafu-u.ac.jp

非常に短い場合があり、待ち時間の間に次のタップが行われた場合、そのタップはタップと判定されなくなってしまう。以上より、ノイズ割合の増加と高速な連続タップの混入により判定精度の低下が起こってしまう問題がある。

ユーザの行動認識手法について、スマートフォンなどのモバイル機器に搭載されているセンサから得られるデータを使って行動認識する手法がある [7][8]。その中で加速度センサとマイクのみを利用してユーザの状態を推定する手法 [9] と同様に、加速度と音を組み合わせることでタップを判定する手法 TIAAS を我々は提案した [10]。しかし、この手法はリアルタイムにタップ判定を行うことができない。

そこで本研究では、TIAAS をリアルタイムにタップ判定できるように改良し、その手法で通常のシングルタップとダブルタップを判定するスマートフォンアプリを開発した。実験では、机に置いたタブレットで z 軸加速度（机に対する上下方向の加速度）と音を測定し、加速度センサを直接タップした時の加速度を使って判定する既存の手法 [6] とアプリの2つを使ってシングルタップとダブルタップの判定を行った。結果は、シングルタップとダブルタップの両方で既存の手法より開発したアプリの方が精度よく判定することができた。

2. 関連研究

2.1 既存のタップ判定手法

既存のタップ判定手法 [6] について説明する。加速度センサを直接2回タップしたことを判定するための手法である。ある時刻の加速度値が閾値 (a_{THT} とする) を越えた時、タップ可能性があるとする。その後、加速度が a_{THT} を越えた時間が一定時間 (T_{DUR} とする) 以下であるとき、タップが行われたと判定する。タップと判定された後は一定の待ち時間 (T_{LAT} とする) の間は次のタップ判定は行わず、待ち時間が終わると次のタップ可能性の判定が可能になるとする。

図1は机に置いた加速度センサで机を1回タップした加速度を測定したときの横軸が時間、縦軸が加速度のグラフである。これを使い動作例を説明する。

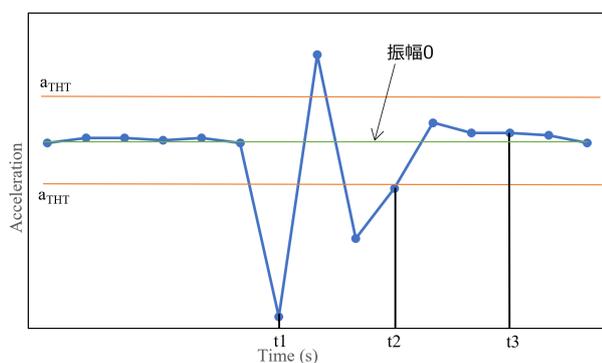


図1 既存の手法の動作例

図1において、グラフ内の点どうしの時間の間隔は0.01 s とし、仮に T_{LAT} を0.06 s とする。時刻 t_1 に加速度が閾値 a_{THT} を越えているため、タップ可能性があるとする。その後加速度が閾値を越えた時間は $t_2 - t_1$ であるので、この時間が T_{DUR} 以下であれば時刻 t_1 にタップが行われたと判定される。時刻 t_1 から T_{LAT} の時間は次のタップ判定は行わないので、図では t_3 の時刻までは次のタップ判定は行われないうこととなる。

既存の手法で T_{LAT} が長い時、 T_{LAT} 内に次のタップが行われた時絶対にそのタップは判定されない。60fpsの音楽ゲームなら連続したフレームにノートが来たとき、ノーツの間隔は0.016 s となる。全てのタップを判定したければ、 T_{LAT} は0.016 s より短くする必要がある。しかし、机を叩いた時入力波の振幅は正か負の方向に振れた後、振動の影響で逆方向にも振れるため、 T_{LAT} を短くすると振動の影響が残っている時刻にも加速度が閾値を越えてしまい、タップと判定されてしまう。

2.2 加速度と音を組み合わせたタップ判定手法

我々が以前提案したタップ判定手法である TIAAS について説明する。TIAAS は机へのタップの際の z 軸加速度と音の振幅を机に置いたタブレットによりタップを判定する。TIAAS ではまずは z 軸加速度のデータを線形補間法によりリサンプリングして不規則にサンプリングされたデータを等間隔にし、 z 軸加速度の振幅の閾値を $a (> 0)$ として閾値を越えたときにタップと判定する。ただし、その直前 k 秒以内にタップが行われたと判定されている場合は、以前のタップの振動が消失しておらず、該当時刻にタップされていない可能性がある。そのため、前回のタップ時刻の z 軸加速度の振幅の大きさ w_{last} に対する現在の z 軸加速度の振幅の大きさ $w_{current}$ の割合を式 (1) により計算する。

$$\text{割合} = \frac{w_{current}}{w_{last}} \times 100 \quad (1)$$

ここで、減衰係数をタップ時 z 軸加速度の振幅が最大値から時間経過でどの程度の値に減衰していくかの割合と定義し、式 (1) により計算した割合が事前に設定した z 軸加速度の振幅の減衰係数より小さい場合はタップされていないとの判定に変更する。式 (1) により計算した割合が事前に設定した z 軸加速度の振幅の減衰係数より大きい場合はタップの可能性があるとし、その後 d 秒内で音の振幅が閾値 $c (> 0)$ を越えている時刻がある場合のみタップと判定する。タップ時の z 軸加速度を表した図2を使って減衰係数について説明する。

図2では時刻 t_1 に z 軸加速度の振幅の大きさが最大となり、 $-1.048 - (-1.332) = 0.284$ となる。この大きさを100%とする。時刻 t_2, t_3, t_4 の z 軸加速度の振幅の大きさはそれぞれ0.132, 0.160, 0.080 となるため、減衰係数は46%, 56%, 28%と計算できる。

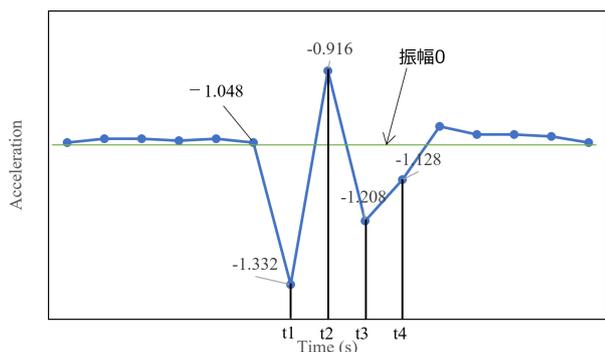


図 2 減衰係数の計算

3. 提案手法

TIAAS は事前に測定した加速度をリサンプリングするという方法を取っているため、リアルタイムにタップ判定ができない。加速度をリサンプリングする必要があった理由は、減衰係数により加速度値の大きさが予測される時刻が限定されており、それらはタップと判定された時刻から 0.063 s, 0.125 s, 0.188 s, 0.250 s, 0.313 s, 0.375 s 後のみであった。そのため、加速度値の取得時刻をその時刻に合わなければいけないからである。そこで、リサンプリングの必要がないように TIAAS で使用した減衰係数の設定のための予備実験から得られたデータより、本研究で使用する減衰係数を式 (2) とする。

$$f(t) = 1.0394e^{-6.4395t} \quad (2)$$

ここで、 t は前回のタップ時刻からの経過時間 (s) とする。TIAAS と同様に、 z 軸加速度の振幅の閾値を $a (> 0)$ としして閾値を越えたときにタップと判定する。ただし、タップ判定時刻の直前 k ms 以内にタップが行われたと判定されている場合は、前回のタップ時刻の z 軸加速度の振幅の大きさ w_{last} に対する現在の z 軸加速度の振幅の大きさ $w_{current}$ の割合を式 (1) により計算する。式 (1) により計算した割合が式 (2) により計算される減衰係数より大きい場合はタップの可能性があるとする。しかし、タップが行われた瞬間より後に加速度値がさらに大きくなる場合が存在する。実際のタップの衝撃による加速度の極大値 (極小値) の直後に正負逆方向にさらに大きな加速度値が計測される場合が多い。このとき上記の手法では実際に行われたタップの回数より多い回数を判定する場合がある。これを防ぐため、タップの可能性があると判定された時刻 t_i の次にサンプリングされた加速度 $a_{t_{i+1}}$ に対しては、時刻 u に計測された加速度値 a_{t_i} と正負が逆の場合、時刻 t_{i+1} ではタップの可能性があると判定しない。例えば図 2 では、加速度の振幅 0 を表す線より加速度値が大きいときは正の方向に、小さい時は負の方向に振れていることを意味する。もし時刻 t_1 の加速度値よりも時刻 t_2 の加速度値の方が大

きかっとしても、時刻 t_2 ではタップの可能性があると判定しない。

次に、音による判定について、音の振幅が閾値 $c (> 0)$ を越えている場合はタップの可能性があるとする。ただし、強いタップが行われた場合は実際にタップが行われた時刻の次にサンプリングされた音の振幅値も閾値を超えることがあるため、音の振幅が閾値よりも大きく、さらに直前にサンプリングされた音の振幅より大きい場合のみタップの可能性があると判定する。最終的に、加速度によりタップの可能性があると判定された後 d 秒内に音によりタップの可能性があると判定されたときのみタップが行われたと判定する。

4. 実験

4.1 実験方法

実験で使うスマートフォンは AQUOS R(SHV39) である。タップの強さを毎回同じにするために、硬式テニスのボールをテーブルから一定の高さから落とすことで人によるタップの代わりとする。図 3 のようにスマートフォンをボール落下地点から 30 cm の位置に置き、スマートフォンで z 軸加速度と音を測定する。 z 軸加速度は 20 ms 間隔でほぼ規則的にサンプリングされ、音は 20 ms から 80 ms 間隔で不規則にサンプリングされる。スマートフォンは提案手法によりタップ判定を行うスマートフォンアプリによりリアルタイムにタップ判定を行う。また、測定した加速度と音の振幅データを出力し、そのデータで既存の手法によりタップ判定を行う。提案手法の a と c と d は事前の実験により $a = 0.2$, $c = 800$, $d = 80$ ms, $k = 150$ ms と設定した。既存の手法の T_{LAT} は筆者がスマホの画面にダブルタップしたときの最短の間隔が 43 ms だったため、43 ms と設定した。

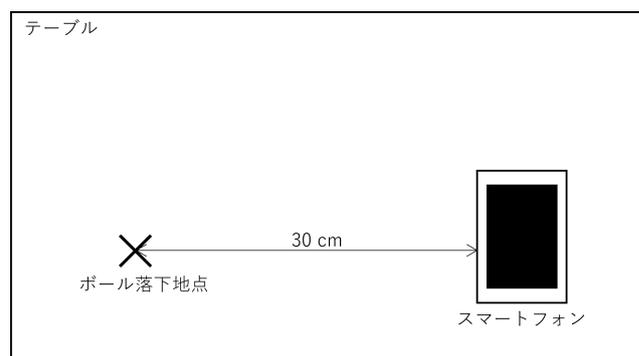


図 3 実験時のスマートフォンの配置

4.2 タップ判定回数の実験

タップが行われたとき、加速度、音の両方でタップの影響がなくなるまで時間がかかる。この残響部分はタップと判定されず、実際のタップ時刻のみタップと判定されな

ればならない. この実験では残響部分に誤ってタップと判定されていないかを調べる. ボールを落とす高さを 30 cm, 20 cm, 10 cm とし, それぞれの高さから 20 回ずつボールを落とし, タップ判定を行った. タップと判定された回数が, 実際に叩いた回数である 1 回なら成功, それ以外なら失敗として成功率を出して, アルゴリズムの性能を評価する.

4.3 タップタイミングの実験

この実験ではダブルタップを行い, タップと判定された時刻と実際のタップ時刻の差を調べる. このとき, 2 回のタップについてタップと判定された時刻と実際のタップ時刻の差が 40 ms 以内なら成功, それ以外なら失敗として 1 回目, 2 回目ののタップについての成功率を出す. ボールを落とす高さを 30 cm, 20 cm, 10 cm とし, それぞれの高さから 2 個ボールを落とすという操作を 20 回ずつ行う. ボールを落とす高さは 2 個とも同じ高さとし, 43-150 ms の時間差で落とすとする.

5. 結果

5.1 タップ判定回数の実験

既存の手法 [6] によりタップの代わりとなるボール落下を 1 回行い, タップと判定された回数を調査した. 60 試行の結果を表 1 に示す. 閾値 a_{THT} は成功率が一番高くなったときの $a_{THT} = 0.56$ である.

表 1 既存の手法がタップと判定した回数

タップと判定した回数	0 回	1 回	2 回
30 cm	1	13	6
20 cm	4	16	0
10 cm	6	14	0

このときの成功率は 72% である. 30 cm の高さからボールを落としたときの加速度の振れは大きいため, a_{THT} を小さく設定すると残響部分に複数回タップの判定が誤って行われる. それを防ぐために a_{THT} を大きく設定すると, 20 cm, 10 cm の高さからボールを落としたときの加速度の振れは 30 cm のときより小さいため, 1 回もタップと判定されないときがあった. このように, 既存の手法は強さの違うタップに対応できていない.

提案手法を実装したアプリにより既存手法と同様にボール落下を 1 回行い, タップと判定された回数を調査した. 60 試行の結果を表 2 に示す.

表 2 開発したアプリがタップと判定した回数

タップと判定した回数	0 回	1 回	2 回
30 cm	0	20	0
20 cm	0	20	0
10 cm	1	19	0

このときの成功率は 98% である. 10 cm の高さからボールを落としたときにタップと判定されなかったケースが 1 回存在したが, それ以外の 59 回はボールを落とす高さによらず残響部分を誤って判定せずにボール落下を判定できている.

5.2 タップタイミングの実験

既存の手法によりダブルタップの判定をしたときの成功率, 成功したタップについてタップと判定された時刻と実際のタップ時刻の差に関して, 20 試行の平均を表 3 に示す.

表 3 既存の手法の成功率, タップと判定された時刻と実際のタップ時刻の差

	1 回目のタップ		2 回目のタップ	
	成功率 (%)	時間差 (ms)	成功率 (%)	時間差 (ms)
30 cm	70	25	35	29
20 cm	85	20	45	24
10 cm	50	17	20	13

提案手法を実装したアプリによりダブルタップの判定をしたときの成功率, 成功したタップについてタップと判定された時刻と実際のタップ時刻の差に関して, 20 試行の平均を表 4 に示す.

表 4 提案手法の成功率, タップと判定された時刻と実際のタップ時刻の差

	1 回目のタップ		2 回目のタップ	
	成功率 (%)	時間差 (ms)	成功率 (%)	時間差 (ms)
30 cm	85	13	60	18
20 cm	95	19	70	14
10 cm	95	14	50	19

成功率について, 既存の手法, 提案手法のどちらも 1 回目より 2 回目のタップの方が低くなった. また, 全てのボール落下の高さにおいて, 1 回目と 2 回目のタップの両方で既存の手法より開発したアプリの方が成功率を高くすることができた. タップと判定された時刻と実際にタップが行われた時刻の差について, 既存の手法, 開発したアプリのどちらも 1 回目と 2 回目のタップで大きな違いはなかった. 10 cm の高さから 2 個目のボールを落としたときの時間差以外は開発したアプリの方が時間差は小さいが, 大きな差はつかなかった.

6. まとめ

机に置いたスマートフォンを使用して既存の方法と開発したアプリでタップ判定を行った. 実験の結果は, 既存の手法より開発したアプリの方がタップの残響部分を誤ってタップ判定する回数少なくすることができ, タップの強さに関わらずタップを判定することができた. 特に, シング

ルタップでは提案手法が98%の高い精度を達成することを確認した。ダブルタップの判定では、タップの強さや最初のタップか2回目のタップかに関わらず既存の手法より開発したアプリの方が精度よく判定することができた。

参考文献

- [1] 則枝真, 三橋秀男, 佐藤誠, "FingerKeypad: 衝撃検出による指上タップ位置の識別," ヒューマンインタフェース学会論文誌 14(4), pp. 393-402, 2012
- [2] Chris Harrison, Desney Tan and Dan Morris, "Skinput: Appropriating the Body as an Input Surface," CHI'10, pp. 453-462, 2010
- [3] 山元亮典, 橋本周司, 三輪貴信, ギエルモエンリケズ, フェイイーヤップ, "タップダンサーの技術向上のためのリズム計測タップシューズの開発," 第78回全国大会講演論文集 2016(1), pp.255-256, 2016
- [4] 伊藤篤志, 清原良三, "加速度センサを用いた二輪車用スマートフォン UI," 研究報告高度交通システムとスマートコミュニティ (ITS) 2015-ITS-61(11), pp. 1-8, 2015
- [5] 池上翔太, 石崎新, 山辺教智, 北上慎二, 梶並知記, 清原良三, "スマートフォンカーナビ向け加速度センサデバイス," 研究報告デジタルコンテンツクリエーション (DCC) 2014-DCC-6(1), pp.1-7, 2014
- [6] モノワイヤレス株式会社." 加速度センサー無線タグ TWELITE 2525A-トワイライトニコニコ".MONO-WIRELESS.COM.
(入手先 (<https://mono-wireless.com/jp/products/TWE-Lite-2525A/index.html>))
- [7] Toshiaki Iso and Kenichi Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," MobileHCI'06, pp. 141-144, 2006
- [8] 池谷直紀, 菊池匡晃, 長健太, 服部正典, "3軸加速度センサを用いた移動状況推定方式," 情報処理学会研究報告ユビキタスコンピューティングシステム (UBI) 2008-UBI-019, pp. 75-80, 2008
- [9] 大内一成, 土井美和子, "携帯電話搭載センサによるリアルタイム生活行動認識システム," 情報処理学会論文誌 Vol.53 No.7, pp.1675-1686, 2012
- [10] Yuika Katayama and Ryo Katsuma, "Improving Accuracy of High-Speed Continuous Tap Recognition on Desk Using Acceleration and Sound Amplitude," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) 2021, pp. 1-8, 2021