Vol.2014-MUS-104 No.5
2014/8/25

ゲームの直感的なインターフェースに向けたリアルタイムでの打音検出 -音のボタン-

安部 武宏[1,a)]　佐古 淳

概要：本稿では，事前に登録した打音の励起をリアルタイムで検出する技術について述べる．ゲームのインターフェースの入力として打音を利用することは，従来のボタンなどの物理的な装置を介した入力よりも直感的であり，音声マイクが低コストかつ手軽に使用できるという点においても有用である．ユーザにストレスを感じさせないための課題として，リアルタイム性，安定した認識率，豊富な登録数，雑音への対処がある．これら課題に対するアプローチとして，打音検出を前提とする処理過程を導入した拡張 NMF(Non-negative Matrix Factorization) を用いる．タブレットデバイス上でリアルタイムで動作できる実験条件にて試行した評価実験では，課題の中で最も重要だと思われる認識率に着目して評価した．最後に，評価の結果を踏まえて本手法のゲームでの実用性について述べる．

1. はじめに

ゲームのインターフェースの入力として使用できる人間の直感的な動作のひとつに，打つ動作を挙げられる．励振によって打音を出すことは音声と並んで直感的であるため，これを利用した打楽器は最も原始的な楽器といわれている．複雑な音色変化やリズムパターンを要求しなければ，打楽器を演奏することは容易である．マレットなどの例外を除いて，多くの打楽器は音高の操作を必要とせず，弦楽器や吹奏楽器と比べて発音することに高い演奏技術を必要としない．手を楽器とみなせば，手拍子でリズムを取ることは誰でも一度はやったことのある打楽器の演奏行為であり，歌声と並んで最も一般的な楽器といえる．

ゲームのインターフェースの入力に打音を使用するにあたって解決が必要な課題を，以下に列挙する．

(1) **リアルタイム性**: ユーザの入力に対して即座にフィードバックされることは必要条件である．このとき，生じる遅延もできるだけ短いほうが望ましい．
(2) **安定した認識率**: ユーザにストレスを感じさせない十分な認識率が必要である．ユーザや励振する物体に依存せずに安定して認識することも重要である．
(3) **豊富な登録数**: 登録できる打音の種類の数は多いほうが望ましい．ゲームにおける入力ボタンの数がこれにあたる．ゲーム中で使用する入力ボタンの数はゲームをデザインする開発者が決めるべきである．
(4) **雑音への対処**: ゲームにおいて BGM と効果音は欠かせない．状況によってはユーザや周りの人々が発する音声の音圧レベルが高くなる．これらは雑音に区分され，打音の認識を阻害する．

これらの課題は全てユーザにストレスを感じさせないことを基準としている．ゲームデザインによっては課題の優先度は前後する．例えば，(4) 雑音への対処においては，自室でひとりで遊ぶジャンルのゲームでは優先度は下がるが，みんなで遊ぶジャンルのゲームでは優先度は上がる．(3) 豊富な登録数においてはその逆もまた然りである．すなわち，ゲームのジャンルやデザインによって課題による制約を緩和することができる．一方，課題を満たせないほどゲームのインターフェースとして使用できる場面が限られる．

我々はゲームのインターフェースに向けた打音検出へのアプローチとして，打音検出固有の課題に対して更新過程を改良した拡張 NMF を用いる．NMF はコスト関数の設計の自由度が高く，簡単に効率的な更新アルゴリズムを実装できるため，様々なゲームデザインで要求される打音検出の認識率や登録数に合わせて拡張できる点で有用である．山本 [1] は打音や噪音の分析に NMF を使用しており，打音検出のアプローチが類似している点で，本研究に最も関連性の高い研究であるといえる．山本は電子楽器のインターフェースの入力として打音や噪音を使用することを目的としており，この場合，楽器に表情をつけて駆動させるために出力は連続値であることが重要である．一方，我々の目的はゲームのインターフェースの入力として打音を使用することであり，打音の有無と音圧レベルの大小がわかる程度の離散値の出力でも構わない．本手法では，打音の正確な周波数成分の分解よりも打音を検出することに重点をおいて，NMF の過程に応じた更新の促進と，特徴点への適応を強くする重み付けを行う．さらに，計算量の削減と雑音への頑健性を高めるために，[2] の知見を参考に周波数成分の特徴的な成分だけを抽出する特徴パスフィルタを登録サンプルから設計し，入力される音響信号への事前処理に用いる．

[1] 無所属
[a)] portnoy1207(at)gmail.com

2. 関連研究

山本は，打音や噪音を入力した電子楽器のインターフェース [1] を提案している．山本の手法とは異なるが，打音検出を利用した電子楽器のアプリケーション [3] が既に商用利用されている．打音のような可聴域の突発的な音以外にも，音や振動をセンシングの対象とした入力インターフェースが提案されている．一例として，腕を叩いたことで生ずる皮膚の振動と骨振動に着目して人体の一部を入力ボタンのように扱える技術 [4] や，物体を伝わる可聴域以上の音を周波数分析してその物体が触れられている状態を推定する技術 [5] などが挙げられる．

音響信号処理固有の課題に対する拡張がされた NMF の一例として，振幅だけでなく位相も分解の対象に含めた複素 NMF[6]，音色をモデル化して組み込んだ NMF[7]，2 つの NMF で打楽器の基底を共有して分解の効率を上げた NMPCF [8] などが挙げられる．また，周波数成分を更新過程で分解するという点で，楽曲中に含まれる打楽器音の発音時刻を検出する手法 [9]，楽譜情報を基に楽器音が潜在する正確な周波数成分を分析する手法 [10] なども技術的に NMF に近い．これら拡張 NMF はリアルタイムでの動作は考慮されていないため，そのまま利用することは難しいが，楽器音の周波数成分の特徴を高度な理論によって分析している点において，参考にする価値は大いにある．

3. 処理の流れ

本章では，ゲームのインターフェースに向けたリアルタイムでの打音検出手法の処理の流れについて述べる．本手法の処理は大まかに登録フェイズと認識フェイズに分かれる．登録フェイズでは，検出を所望する打音をその場でマイクから入力して，後に説明する拡張 NMF の基底となるテンプレートを生成する．認識フェイズでは，拡張 NMF を用いて，マイクから入力された音響信号の周波数成分からテンプレート強度へと分解し，テンプレート強度の時系列から励起を検出する．登録フェイズ，処理フェイズに関わらず，全ての音響信号はマイクから入力され，観測スペクトルに変換される．処理の概要を**図 1** に示す．

3.1 観測スペクトル

マイクから次々と入力される音響信号をフーリエ変換することによって振幅スペクトル (観測スペクトルと呼ぶ) を分析する．本手法では区間の境界近辺におけるデータの損失を避けるために，音響信号に窓関数を積算しない．打音は音圧レベルが瞬時に増幅して減衰する特徴を持っているため，この損失によって励起そのものが失われる恐れがある．この損失は，フレーム長に対して十分にシフト幅を短くにすることで補うことができるが，シフト幅に反比例して分析する音響信号の総量が増える．本手法はリアルタイムでの動作が必要条件であり，シフト幅を十分に短くすることが難しいため，窓関数を使用しないことでデータの損失自体を回避している．これにより，本来音響信号に含まない周波数成分が含まれて若干の歪みが生じるが，離散値

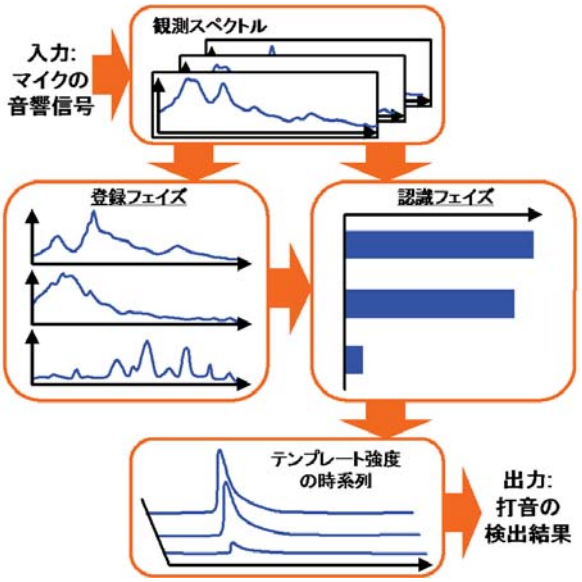


図 1 本手法の概要

を出力とする打音検出においてこの影響は小さい．

また，音響信号処理に NMF を用いる場合，しばしば時系列に並んだ複数のフレームの周波数成分や特徴量を基底とするが，本手法では以下の観点より，1 フレームの観測スペクトルのみを基底としている．

(1) **検出結果に瞬時性を持たせる**
基底に用いるデータ区間の長さは直結して遅延となるため，データ区間は短いほうが望ましい．

(2) **打音の時系列における周波数成分の変化**
打音は励起時にエネルギーが集約されるため，励起部分だけを基底にしても打音検出は可能である．

(3) **計算量の削減**
基底の削減は NMF の更新過程における計算量の削減へとつながる．

3.2 登録フェイズ

登録フェイズにおいては，登録を所望した打音のみがマイク入力されると仮定しているため，簡単な閾値処理で打音を検出する．具体的には次々と得られる観測スペクトルの周波数方向の合算値をエネルギーとして算出し，時系列のエネルギー間の差分を閾値処理することで打音の励起を検出する．打音の励起部の観測スペクトルを登録サンプルとして集め，一定数が集まったところで，登録サンプルの中央値を得る．本稿ではこの中央値をテンプレート $T_k(f)$ と呼ぶ．ここで，k と f はそれぞれ，楽器の種類と周波数成分における周波数軸である．

この時点で得られたテンプレートをそのまま，NMF の基底として用いても構わないが，本手法では認識率の向上のために特徴点，計算量の削減のために特徴パスフィルタをそれぞれ，登録サンプルから導出し，これらに基づいて特徴的なテンプレートの成分のみを選出する．

3.2.1 周波数成分の特徴点

一定強度を持つ上位数点の周波数成分のピークを特徴点と定義する．一定強度を持つが下位のピークは特徴点から棄却される．打音の周波数成分に特徴点を付け加えた概要を**図 2** に示す．特徴点 f_c とその集合 F_C を次式で表す．

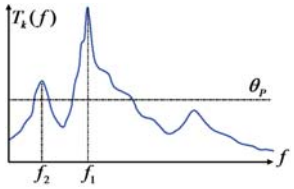


図 2 特徴点の付与．本図では一定強度 θ_P を持つピーク f_1 と f_2 が特徴点となる

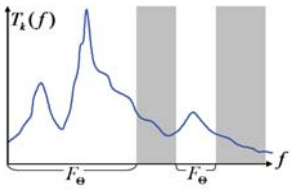


図 3 特徴パスフィルタ．灰色の領域はフィルタリングにより棄却される

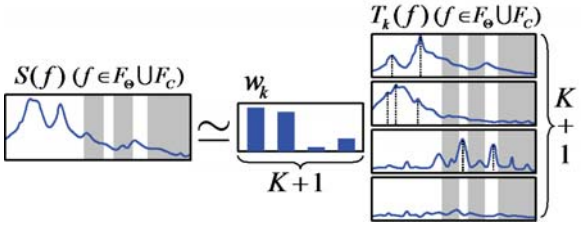


図 4 本手法で用いる NMF の概要．灰色の領域は基底に含まれない

$$\{f_c(c=1,2,\cdots)\} \in F_C \quad (1)$$

特徴点においては新たに重み付けされた距離が NMF のコスト関数に上乗せされる．

3.2.2 特徴パスフィルタの設計

特徴パスフィルタは登録サンプル間の周波数成分の分布に基いて導出される．本手法では特徴パスフィルタを次式の特徴評価関数 $I(f)$ に基いて設計する．

$$I(f)=(F_{nq}-f)\sum_k^K T_k(f)T_{k\max}(f)T_{k\min}(f)V_k(f) \quad (2)$$

ここで，F_{nq} はナイキスト周波数，K は登録した打音の種類の総数，$T_k(f)$，$T_{k\,\max}(f)$，$T_{k\,\min}(f)$，$V_k(f)$ はそれぞれ，登録サンプル間における中央値，最大値，最小値，分散である．特徴評価関数は低周波数領域，高い強度が分布している周波数領域，登録サンプル間の強度が分散している周波数領域において高い値になる．特徴パスフィルタは，次式のような特徴評価関数の値が上位の周波数領域の集合のみを通過させる．

$$\{f|I(f) > I_\Theta\} \in F_\Theta \quad (3)$$

ここで，I_Θ はフィルタリングの割合を決めるための閾値であり，特徴評価関数が算出される度に決定される．特徴パスフィルタによるフィルタリングの概要を**図 3** に示す．

3.3 認識フェイズ

拡張 NMF を用いて，マイク入力された観測スペクトルと登録フェイズで算出したテンプレートから観測スペクトルに含まれるテンプレートの重み，すなわちテンプレート強度を導出する．テンプレート強度はマイクから音響信号が入力される度に算出され，最後に，時系列に並んだテンプレート強度を閾値処理することで打音を検出する．

3.3.1 拡張 NMF からのテンプレート強度の導出

本手法では，効果的に打音を検出するために，コスト関数の設計と基底の更新過程を拡張した NMF を用いる．観測スペクトルとテンプレートは特徴パスフィルタを通過した部分のみを用いる．基底となるテンプレートは登録した各打音のテンプレートの他に，雑音成分を表現するテンプレートを含む．以下，では前者を打音テンプレート，後者を雑音テンプレートと呼ぶ．本手法で用いる NMF における観測スペクトル $S(f)$，テンプレート強度 w_k，打音テンプレート及び雑音テンプレート $T_k(f)$，の位置づけを**図 4** に示す．このとき，基底の数は打音テンプレートに雑音テンプレートを足した $K+1$ となる．また，観測スペクトルと各テンプレートは $F_\Theta \cup F_C$ に属する領域のみ分解に用いられる．NMF の更新時には登録フェイズで得られたテンプレートは更新せず固定し，テンプレート強度，雑音テンプレートを交互に一定回数更新し，最後に更新された値を収束値として使用する．

NMF のコスト関数を設計するにあたって広く用いられる距離として，ユークリッド距離の二乗誤差，KL ダイバージェンス，IS ダイバージェンスなどが挙げられる [11]．本手法の NMF ではユークリッド距離の二乗誤差に基づいてコスト関数を設計する．テンプレートとテンプレート強度の積 ($M(f)=\sum_K w_k T_k(f)$，重畳テンプレートと呼ぶ) が観測スペクトルよりも小さい場合，ユークリッド距離の二乗誤差では重畳テンプレートの適応が比較的緩やかである．音源分離などでは，重畳テンプレートの適応が促進されるように KL ダイバージェンスや IS ダイバージェンスがしばしば用いられる．一方，打音検出においてこの促進は所望していない打音が検出されるといった誤認識につながる．このため，予備実験ではユークリッド距離の二乗誤差を適用したときが最も良い認識率となった．

テンプレート強度は特徴点への適応が重み付けされて算出される．具体的には特徴点において観測スペクトルに重畳テンプレートが適応しやすくする距離をコスト関数に加算する．ユークリッド距離の二乗誤差に基いたコスト関数から導出されたテンプレート強度の更新式を以下に示す．

$$w_k \leftarrow w_k \frac{\int_{F_\Theta} S(f)T_k(f)df + \alpha\int_{F_C} S(f)^2 T_k(f)df}{\int_{F_\Theta} M(f)T_k(f)df + \alpha\int_{F_C} M(f)^2 T_k(f)df} \quad (4)$$

ここで，α は特徴点への適応に関する重みである．α を大きくするほど，重畳テンプレートの特徴点の成分が観測スペクトルへ強く適応される．

本手法では，ゲームのインターフェースの入力として打音を用いることを目的としているため，最終的に得られるテンプレート強度は，打音の正確な音量を表現した連続値ではなく，打音の大まかな音量及び有無を表現した離散値でも構わない．そこで，更新過程の中盤にて，更新中のテンプレート強度が大きい場合はさらに大きくして，小さい場合はゼロに近づける更新を挟む．具体的には (4) 式を用いて更新されたテンプレート強度を次式を用いてさらに更新する．

$$w_k \leftarrow \begin{cases} w_k \cdot (1-\beta_d \mathrm{N}(r, R/2, \sigma^2)) & (w_k < W_d) \\ w_k \cdot (1+\beta_b \mathrm{N}(r, R/2, \sigma^2)) & (W_b < w_k) \end{cases} \quad (5)$$

ここで，r は現更新回数，R は更新回数の合計，$\mathrm{N}(x,\mu,\sigma^2)$ はガウス分布 (x は変数，μ は平均，σ^2 は分散) 上の 1 点，β_d と β_b はそれぞれ，大きな値への更新と小さな値への更新を促進する重み，W_d と W_b はそれぞれ，大きな値への更新と小さな値への更新に関する境界値である．これにより，少ない更新回数でテンプレート強度が収束し，さらに，検出を所望しないテンプレートの誤った適応によって生じ

る微小なテンプレート強度がゼロへ収束し，検出を所望する打音テンプレートの適応が正確になる．

雑音によって重畳テンプレートの観測スペクトルへの適応が阻害されるのを軽減させるため，雑音テンプレートをNMF の基底に追加する．打音テンプレートと異なり，雑音テンプレートは更新対象に含まれ，すなわち，観測スペクトルと重畳テンプレートに応じて柔軟に形を変える．雑音テンプレートは次式によって更新する．

$$T_\epsilon(f) \leftarrow T_\epsilon(f)\frac{\int S(f)df}{\int M(f)df} \tag{6}$$

(4)，(5)，(6) を繰り返すことによってテンプレート強度と雑音テンプレートは収束していく．

3.3.2 テンプレート強度の閾値処理

拡張 NMF より得られた時系列のテンプレート強度を閾値処理することによって打音の励振を検出する．0 を検出なし，1 を検出ありとして，次式によってテンプレート強度から打音の励振を検出する．

$$\gamma_k(t)=\begin{cases} 1 & \left(\begin{array}{ll} \text{if} & W_B > w_k(t-\Delta t), \\ \text{and} & W_C < w_k(t), \\ \text{and} & W_D < w_k(t)-w_k(t-\Delta t). \end{array}\right) \\ 0 & (\text{otherwise}) \end{cases} \tag{7}$$

ここで，Δt はシフト幅の倍数からなる微小値で，W_B, W_C, W_D はそれぞれ，過去のテンプレート強度に対する閾値，現在のテンプレート強度に対する閾値，テンプレート強度の増加量に対する閾値である．この数式は，テンプレート強度が十分に小さい値から十分に大きい値に瞬時に増加したときを正とすることを意味している．

さらに，打音検出の状況の仮定を置くことで誤認識を抑える．具体的には，同時発音される上限数は K' 音までという仮定と，各打音ごとの励起に一定の間隔が存在するという仮定に基づき，次式によって検出結果を棄却する．

$$\gamma'_k(t) = \sum_{i=1} \hat{\gamma}_k(t - i\Delta t) \tag{8}$$

$$\psi_k(t)=\begin{cases} 1 & \left(\begin{array}{ll} \text{if} & 0 = \gamma'_k(t), \\ \text{and} & K' > \sum_k^K \gamma'_k(t), \\ \text{and} & W_{K'}(t) \le w_k(t). \end{array}\right) \\ 0 & (\text{otherwise}) \end{cases} \tag{9}$$

$$\hat{\gamma}_k(t) = \gamma_k(t)\psi_k(t) \tag{10}$$

ここで，$W_{K'}(t)$ は，K' 番目に値が大きいテンプレート強度，$\gamma'_k(t)$ は，過去の一定時間における打音の検出の有無，$\hat{\gamma}_k(t)$ は，最終的な検出結果である．

4. 評価実験

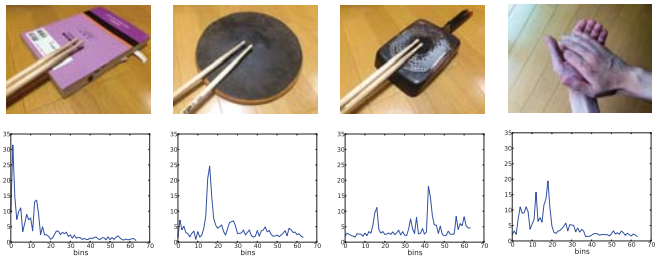
本章では，本手法で用いた拡張 NMF と一般的な NMF の性能を評価するために行った比較実験について報告する．

4.1 実験条件

本実験は据え置きの計算機上でバッチ処理によって試行しているが，本手法はゲームのインターフェースとして利用することを目的としているため，タブレットデバイス上でリアルタイムで動作できるように実験条件を定める．基準とするタブレットデバイスとして，初期型の Nexus7 を採用する．打音を 5 種類登録したときでも，Nexus7 上でリアルタイムで動作することを必要条件として，**表 1** に示すように水準の値を決定した．使用言語は，Android SDK で用いられている Java とした．実験データの収録には実環境での使用を想定して Nexus7 の備え付けマイクを用い，低周波数領域における周波数分解能を稼ぐために，低めのレートでサンプリングした．フレーム長とシフト幅は，初心者及び熟練したドラマーのシングルストロークの打数がおおよそ 1 秒間に 8 打 [12] という知見を参考に，この条件でも 1 打 1 打個々に分析できる長さとした．特徴パスフィルタの通過サンプル数の決定は NMF の処理量を半分 (129 点 [周波数ビン数]×0.5 ≈ 64 点) にすることを目安とした．

表 1 リアルタイムでの動作を可能とする実験条件

水準	値
使用言語	Java
サンプリングレート	11025Hz
周波数分析のフレーム長	256 点
周波数分析のシフト幅	128 点
特徴パスフィルタの通過サンプル数	64 点



a: 本　b: 練習パッド　c: フライパン　d: 手 (手拍子)

図 5 実験データ収録時に用いた道具とその打音のテンプレート (特徴パスフィルタを通過したサンプルのみ) の一例．

実験データの内訳は，ドラムスティックで励振させることによって得られる 4 種類 (本，ドラム用の練習パッド，フライパン，スティック) の打音，手拍子によって得られる 1 種類の打音の合計 5 種類である．各種類ごとに 100 音，すなわち合計 500 音用意する．実験データ収録時に用いた道具とその打音のテンプレートの一例を **図 5** に示す．同時発音時における各打音の種類間の音圧レベルの差をなくすために実験データの励起部の音量と発音時刻を揃えている．

認識率は 10-fold cross validation で算出する．このとき，各種 100 音の内，10 音を登録サンプル，90 音を認識対象とする．各打音の同時発音数の上限は腕の数を考慮して 2 音までとして，5 種類の打音から 1 音及び 2 音を選ぶ組み合わせを考える．すなわち，合計の試行回数は $({}_5C_1 + {}_5C_2) \times 10 \times 90 = 13500$ 回となる．

本実験では，ベースライン手法を拡張のない NMF として，提案した NMF の拡張と組み合わせた．具体的には，ベースライン手法に特徴パスフィルタを組み合わせたものを手法 (i)，手法 (i) に特徴点の利用を組み合わせたものを手法 (i)+(ii)，手法 (i)+(ii) にテンプレート強度の更新の促進を組み合わせたものを手法 (i)+(ii)+(iii) と定義して比較評価した．テンプレート強度の閾値処理に用いる K' は既知として同時発音される打音数を与え，この数より多くの打音は一度に検出されないようにした．また，1 回の試行で同じ打音が 2 回以上検出された場合は誤認識として扱う．

4.2 実験結果

表 2 に各打音の組み合わせごとの認識率を示す．ベースライン手法と比較して本手法の認識率が優位であることがわかる．

ベースライン手法と手法 (i) の認識率に大きな差はみられなかった．これは，特徴パスフィルタを通してデータ列を削減した基底を用いても十分な認識率を得られたことを示している．本実験では特徴パスフィルタの通過率を 50%としている．これは，NMF の計算量が基底のデータ列の長さに応じて線形に増えていくので，計算量を半分にしつつ認識率を維持できたことを意味する．

手法 (i) と手法 (i)+(ii) の比較によって，特徴点の利用が認識率の改善に作用していることがわかる．最も改善したのは 19.2%の練習パッドの打音のみの場合であった．実験データに用いた練習パッドは材質や構造上，励振する打点の位置やスティックの握り方によって音色が変化しやすい性質を持つ．よって，周波数成分全体の分布は安定しないが，特徴点の周波数領域が安定する打音に対して，特徴点の利用は効果があると考えられる．一方，フライパンと手拍子の打音と組み合わせの認識率は −19.0% となり悪化している．フライパンと手拍子はどちらも比較的高い周波数領域に成分が分布する．また，手拍子は叩き方が少しでも変わると周波数成分と特徴点の周波数領域が大きく変わる．これらの要因が重なって，特徴点の利用が逆効果に働いたと考えられる．

手法 (i)+(ii) と手法 (i)+(ii)+(iii) の比較によって，テンプレート強度の更新の促進が全ての打音の認識率の改善に作用していることがわかる．特にフライパンの打音を含む組み合わせの改善が大きく，本とフライパンの打音の組み合わせにおいては認識率が 29.9%改善した．フライパンの打音と練習パッドの打音は低周波数領域における特徴点となりうる周波数成分において類似している．そのため，手法 (i)+(ii) ではフライパンの打音のみを入力してもフライパンと練習パッドのテンプレートに分解されるケースが多く見られた．テンプレート強度の更新の促進によって，観測スペクトルに存在しない打音のテンプレート強度が更新過程の中盤で小さくなり，検出を所望する打音のテンプレートのみに適応されて分解されるようになる．この更新の促進がフライパンの打音の観測スペクトルに練習パッドのテンプレートが適応されることを抑制して，フライパンのテンプレートの適応が良くなったと考えられる．

本と練習パッドの打音の組み合わせが最悪の認識率 (60.6%) となった．誤認識しているケースを確認したところ，本の打音が正しく認識されている一方，手拍子による打音のテンプレートが練習パッドのテンプレートよりも強く適応されて，練習パッドの打音が認識されずに手拍子による打音が認識されていた．同時に発音された本と練習パッドの打音と，手拍子の打音は周波数成分が類似している．各打音 1 音の周波数成分が似ている場合，テンプレート強度の分解がスパースにならないことは自明であるが，2 音を重ねあわせた周波数成分が似ている場合でも分解がスパースになりづらいことを意味している．

5. 入力インターフェースとしての打音検出

本章では，実用性の観点から，インターフェースの入力として打音を適用する上での分野ごとにおける制約や利点について議論する．

5.1 ゲームのインターフェースの入力として

従来のゲームにおいて励振の検出は，入力ボタンやピエゾ素子の付いたパッドなどが用いられてきたが，励振時に発生する打音や振動に着目すれば音声マイクやピエゾマイクを用いて励振を検出することもできる．後者のマイクは前者に比べて，ハードウェアの側面で低コストであり，使用の手軽さで優位である．

打音検出をインターフェースとして利用するにあたっては，ゲームデザインで要求されている認識率と認識結果が出るまでの遅延を把握することが重要である．例えば，音ゲーと呼ばれる音楽と譜面に合わせて入力するゲームにおいては，100%に近い認識率と十分に短い遅延と，両方において高い水準が求められる．一方，画面の中のキャラクターとインタラクションする類の多少の入力誤りが許されるゲームでは，認識率と遅延の両方がある程度の水準を満たしていればよい．

本手法では同時発音数が 1 音の場合，打音によっては 9 割強の十分な認識率を得ることができている．登録する打音の種類を減らしたり，周波数成分が似ていない打音を利用すればさらに認識率は 100%に漸近していくと予想されるため，条件付きでなら音ゲーに利用できる可能性は高い．同時発音数が 2 音の場合，十分な認識率を得られておらず，登録する打音の種類を減らすなどをしても音ゲーが要求する認識率には届かないと考えられる．多少の入力誤りが許されるゲームでなら同時発音数などの条件に関わらず十分に適用できると思われる．赤外線 LED の追従及び加速度センサに基づくモーション認識技術や，カメラからの画像認識技術といった様々な認識技術が，既にゲームのインターフェースとして実用化されている．これら認識技術の全てで 100%の認識率を達成できているとはいいがたいが，ゲームデザインにあった使い方がされているため，ユーザに直感的な入力の楽しさを十分に提供できている．

マイクのバッファサイズが実験条件と同じシフト幅分確保できれば，遅延は 11ms($\approx 128/11025\times1000$) ほどとなる．しかし，実験条件の基準の環境とした Nexus7 を利用した場合，サンプリングレートが 11025Hz のときの最小バッファサイズは 512 点であるため，遅延は 46ms($\approx 512/11025\times1000$) ほどとなる．すなわち，遅延は本手法ではなくマイクのバッファサイズに依存している．ゲームの分野での知見ではないが，熟練した演奏家における楽器音の遅延の感じ方を調査した報告 [13] では，条件によって許容できる遅延は 42ms から 1.4ms 程度だと報告されている．ゲームの遅延の感じ方もこの知見に近いとすると，音ゲーを遊ぶにはギリギリの遅延である．マイクのバッファサイズの制約がなければ十分に短い遅延でゲームができると期待できる．

表 2　各打音の認識率．beseline は拡張なしの NMF．各項目 (i) 特徴パスフィルタ，(ii) 特徴点の利用，(iii) テンプレート強度の更新の促進は NMF への拡張の種類

K'	打音の組み合わせ					NMF の拡張の種類			
	本	練習パッド	フライパン	手拍子	スティック	baseline	(i)	(i)+(ii)	(i)+(ii)+(iii)
1	○	−	−	−	−	100.0%	100.0%	100.0%	100.0%
	−	○	−	−	−	56.6%	62.4%	81.6%	88.4%
	−	−	○	−	−	87.7%	82.0%	81.0%	98.6%
	−	−	−	○	−	97.9%	99.0%	98.6%	99.6%
	−	−	−	−	○	97.1%	96.3%	96.2%	99.3%
2	○	○	−	−	−	35.8%	39.2%	54.4%	60.6%
	○	−	○	−	−	45.4%	53.3%	51.8%	81.7%
	○	−	−	○	−	77.9%	81.6%	76.9%	86.9%
	○	−	−	−	○	97.4%	96.7%	97.2%	98.8%
	−	○	○	−	−	43.9%	43.2%	61.8%	80.2%
	−	○	−	○	−	47.6%	51.7%	67.3%	83.0%
	−	○	−	−	○	64.4%	67.0%	84.7%	92.1%
	−	−	○	○	−	80.6%	80.9%	61.9%	72.4%
	−	−	○	−	○	89.8%	90.8%	90.0%	90.8%
	−	−	−	○	○	97.7%	96.3%	94.4%	96.1%

5.2　その他のインターフェースの入力として

打音検出は電子楽器のトリガーとしても利用できる．電子楽器の場合，認識率よりも遅延の大きさが重要であるが，前節で述べたとおり，マイクのバッファサイズの制約がなければ電子楽器の入力として利用できると期待される．

打音検出は DTM におけるリズムパターン入力と相性が良い．MIDI キーボードにはリズムパターンを打ち込むための小型のドラムパッドが付いているものがあるが，ドラムパッドを叩く感覚が直感的という理由から，これを用いる DTM ユーザもいる．打音検出を使うことであらゆる形の物体をドラムパッドのように扱うことができる．実際の打楽器をある程度演奏できる DTM ユーザにとっては，打音検出がより直感的にリズムパターンを打ち込むためのツールとなり得る．

また，打楽器の教則支援にも打音検出を適用可能である．例えば，提示された練習譜面を雑誌上で正確に演奏できているかを判定するという打楽器の教則支援アプリが実現できる．ダブルストロークを多用した速いリズムパターンなどを練習することは，現時点では認識率や時間分解能の観点から難しいが，教則支援の需要は初心者に多いと考えられるため，致命的な問題ではない．

6.　おわりに

本稿では，ゲームのインターフェースに向けた打音検出手法について述べた．タブレットデバイスという処理能力やマイクの感度が限定的な環境で評価実験を試行した．そのため，処理能力の高い据え置き型の計算機や感度の高いマイクの利用，また処理の早い言語の使用や最適化によって，登録する打音の上限数の増加と僅かながらの実験結果より認識率を改善することが期待できる．

今後は，さらなる認識率の向上及び雑音の対処について検討し，雑音を加算した実験データにて評価実験を行う．また，打音以外の音をゲームのインターフェースとして利用することを検討する．認識対象の例として噪音や音高を持った楽器音などが挙げられる．

謝辞　過去に所属していた研究室の後輩から貴重な意見をいただいた．趣味で進めていた研究をこの場で発表するきっかけとなった．彼らに深謝する．

参考文献

[1] Kazuhiko, Y.: Possessing Drums: An Interface of Musical Instruments that Assigns Arbitrary Timbres to Personal Belongings, 情報処理学会論文誌，Vol. 21, No. 2, pp. 274–282 (2013).
[2] 山本俊一，Valin, J.-M.，中臺一博，中野幹生，辻野広司，駒谷和範，尾形哲也，奥乃　博：音源分離との統合によるミッシングフィーチャマスク自動生成に基づく同時発話音声認識，日本ロボット学会誌，Vol. 25, No. 1, pp. 92–102 (2007).
[3] TableDrum: TableDrum (online), available from ⟨http://www.tabledrum.com/⟩ (accessed 2014-07-30).
[4] Harrison, C., Tan, D. and Morris, D.: Skinput: Appropriating the Body as an Input Surface, *Proc. CHI'10*, pp. 453–462 (2010).
[5] Ono, M., Shizuki, B. and Tanaka, J.: Touch & Activate: Adding Interactivity to Existing Objects Using Active Acoustic Sensing, Proc UIST'13, pp. 31–40 (2013).
[6] 亀岡弘和，小野順貴，嵯峨山茂樹：複素 NMF：新しいスパース信号分解表現と基底系学習アルゴリズム，日本音響学会秋季研究発表会講演集，No. 2-8-13, pp. 657–660 (2008).
[7] 安良岡直希，奥乃　博：調波・非調波・音色構造因子分解による音響信号分析と音源分離インターフェースへの応用，MUS-94, No. 27, pp. 1–8 (2012).
[8] Yoo, J., Kim, M., Kang, K. and Choi, S.: Nonnegative matrix partial co-factorization for drum source separation, *Proc. ICASSP*, IEEE, pp. 1942–1945 (2010).
[9] 吉井和佳，後藤真孝，奥乃　博：テンプレート適応を利用した実世界の音楽音響信号に対するドラムスの音源同定，MUS-56, No. 127, pp. 55–60 (2003).
[10] 糸山克寿，後藤真孝，駒谷和範，尾形哲也，奥乃　博：楽譜情報を援用した多重奏音楽音響信号の音源分離と調波・非調波統合モデルの制約付きパラメータ推定の同時実現，情報処理学会論文誌，Vol. 49, No. 3, pp. 1–10 (2008).
[11] 澤田　宏：非負値行列因子分解 NMF の基礎とデータ/信号解析への応用，電子情報通信学会誌，Vol. 95, No. 9, pp. 829–833 (2012).
[12] Fujii, S., Kudo, K., Ohtsuki, T. and Oda, S.: Tapping performance and underlying wrist muscle activity of non-drummers, drummers, and the world's fastest drummer., *Neuroscience Letters.*, pp. 69–73 (2009).
[13] Lester, M. and Boley, J.: The Effects of Latency on Live Sound Monitoring, AES 123rd Convention (2007).