

Onomatrack: リズムイメージを直接表出することを 目指したオノマトペによるリズムトラック制作システム

有馬 丞^{1,a)} 山本 景子^{1,b)} 倉本 到^{1,c)} 辻野 嘉宏^{1,d)}

概要: 音楽制作において、思いついたリズムパターンを手早く記録したいことがある。そのようなリズムパターンのイメージは、複雑なものであればあるほどすぐに曖昧になってしまうが、既存の記録方法では手早く簡単に記録することが難しい。そこで本稿では、ユーザが思いついたリズムパターンのイメージを直接表出することができるようにすることを目的に、オノマトペを用いることで発音タイミングと音色を指定し、同時にその音量・定位の指定を行う手法を提案する。この手法を用いることで「音のラフスケッチ」としてリズムトラックを手早く制作できる。提案手法に基づき実装したリズムトラック制作支援システム“Onomatrack”を評価した結果、既存のリズムトラック制作環境と比較してイメージしたリズムパターンを手早く表現できる、という評価が得られた。また、リズムトラックに使われる音色をオノマトペで指定することで、既存のリズムトラック制作環境よりも手早く音色の選択ができることと評価された。

1. はじめに

ギターなどの楽器を演奏する人間にとって、普段、何気なくその楽器を演奏していて、気に入ったコード進行を思いつくと同時に、その思いついたコード進行に合ったリズムパターンのイメージを思いつくことがよくある。そして、そのリズムパターンを録音した音楽データであるリズムトラックを手早く作り、それに合わせて演奏することで、思いついたコード進行とリズムパターンの組み合わせの出来を確認したい、ということも多い。

このとき、思いついたリズムパターンのイメージが、ベースドラム、スネアドラム、ハイハットなどの通常のドラムセットに含まれる音だけを使ったもので、しかも単純なリズムパターンの場合は、MIDI キーボードや電子ドラムなどを使って音を入力し、記録することで、手軽にリズムトラックを制作することができる。しかし、1小節あたりの音数が多く、通常のドラムセットに含まれる音以外の、テレビの時報やシンセサイザで作ったような電子的なノイズ音や、ドアが「バタン」と閉まるときの音のような環境音などを含み、個々の音の音量や定位^{*1}が頻繁に変化する複雑なリズムパターンの場合、そのリズムパターンのイメージを

リズムトラックとして手早く制作することは難しい。

なにかを思いついたときのイメージは、普通、アイデアの断片のみからなる曖昧なものである場合が多い。このイメージは時間が経つにつれてさらに曖昧になっていき、しばらくするとそのイメージ自体がどのようなものだったかわからなくなってしまふ。このようなイメージを忘れないうちに記録しておくために、我々は普段、文字でメモをとったり、ラフスケッチとして絵のかたちで描き出す。これはリズムパターンの場合も同様で、新しいリズムパターンを思いついた瞬間からそのイメージはぼやけ始めてしまふ。しかも、これは思いついたリズムパターンのイメージが複雑であるほど顕著である。なぜなら、複雑なリズムパターンのイメージの細部を長時間記憶しておくことは難しいからである。ゆえに、ユーザの思いついたリズムパターンのイメージが曖昧にならないうちに、そのイメージを記録した、いわば「音のラフスケッチ」としてのリズムトラックを、できるだけ手早く制作できることが望まれる。この「音のラフスケッチ」は、最初に思いついたコード進行と組み合わせることでリズムパターンのイメージを思い出せるようなものでなければならない。

従来方法では、複雑なリズムパターンを思いついたときにリズムトラックを制作しようとしても、前述した複雑なリズムパターンの持つ特徴から、その制作自体に時間がかかる。そのため、リズムパターンのイメージが曖昧になってし

¹ 京都工芸繊維大学
Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto, Japan

a) arima@hit.is.kit.ac.jp

b) kei@kit.ac.jp

c) kuramoto@hit.is.kit.ac.jp

d) tsujino@kit.ac.jp

^{*1} ある音が空間中のどの辺り、またはどの方向から聴こえてくるか、という位置。

まい、結局最初に思いついたイメージを表したリズムトラックを制作することができないという問題がある。そしてそのことで、ユーザは音楽制作自体が期待通りにならないために、面倒であると感じ、せつかく思いついたリズムパターンを活かした曲作り自体を諦めてしまう可能性がある。

そこで、思いついたリズムパターのイメージを直接表出した「音のラフスケッチ」を手早く制作できるようにすることを目指し、オノマトペを用いて発音タイミングと音色を指定し、同時にその音量・定位の指定を行うことで、ユーザのイメージに近いリズムパターンを表現したリズムトラックを手早く制作する手法を提案する。

2. リズムパターの記録

2.1 リズムパターと「音のラフスケッチ」

ユーザがイメージするリズムパターンを表現するためには、以下のパラメータを記録する必要がある。

- 音色：リズムパターンに使われているある音がどのような音色を持っているか
- 発音タイミング：リズムパターンに使われているある音がどのタイミングで鳴るか
- 音量：リズムパターンに使われているある音がどの程度の大きさで鳴るか
- 定位：リズムパターンに使われているある音が空間中のどの位置で鳴るか

これらのパラメータは簡単なリズムパターンを表現する際にも必要であるが、複雑なリズムパターンを表現する場合は、音色、発音タイミングが記録できることに加えて、特に音量、定位をユーザのイメージに近いかたちで記録する必要がある。また、思いついたリズムパターンを活かした音楽制作をより手軽に行うためには、記録したリズムトラックと合わせて演奏し、アイデアの出来がすぐに確認できるようにすることが必要である。

よって、以下の条件を満たす「音のラフスケッチ」としてのリズムトラックを制作する支援が必要である。

- 最初に思いついたリズムパターのイメージが曖昧にならないうちに、手早く記録できること
- 思いついたアイデアの出来を確認するために、その場で再生できるリズムトラックとして記録されていること
- 後から確認したときに、思いついたリズムパターのイメージを思い出せるだけの情報として、前述の4種のパラメータが記録されていること

2.2 既存の記録方法

ユーザがイメージするリズムパターンを記録する方法として古くから用いられている方法には、

- (1) 声や楽器の演奏を録音して記録する方法
- (2) 絵・記号などを用いて記録する方法

がある。

(1) は、ユーザがイメージするリズムパターンが簡単なものの場合、最も手早く直接的に記録でき、再生も容易である。しかし、複雑なリズムパターの場合、複数の音を同時に表現することや非常に短い間隔で連続的に発音される音や定位の変化を表現することが困難である。また、ユーザがイメージする音を作り出すことに習熟しなければならない。これを解決するシステムとして、中野らの Voice Drummer[1] や、Apple 社の GarageBand[2] などのループシーケンサが挙げられる。しかしこれらの支援システムを用いても、

- きわめて短い間隔で連続して発音される音を表現することが困難である
- 定位は表現できない
- 音楽素材を探す手間がかかる
- 個々の音のパラメータの指定に手間がかかる

のいずれかの問題が残されるため、複雑なリズムパターの記録には適していない。

それに対し (2) は、ユーザの思いついたイメージの自由度を保ったまま記録することが可能である。しかし、ユーザがなんらかの表記法を確立したり、表記法に習熟する必要がある。また、記録されたものを再生するまでにタイムラグがあるという問題がある。これを解決するシステムとして、Farbood らの Hyperscore[3] や Forsberg らの The Music Notepad[4] が挙げられる。しかし、

- システム固有の表記法を学習する必要がある
- 環境音や電子的ノイズ音は使用できない
- 定位は表現できない

という問題が残る。

現在リズムトラック制作には、以上の問題を比較的解決する DAW (Digital Audio Workstation) 環境が使用されることが一般的である。しかし以下に挙げるような手間があり、手早く制作することが難しい。

- (a) ドラムセットの音に含まれないような音色は、ユーザのイメージに近い音を探さなければならず、手間がかかる
- (b) ユーザがイメージする発音タイミングに音を配置するのに手間がかかる
- (c) リズムトラックに含まれる多くの音のそれぞれに対して音量や定位を決める必要があり、これらの指定に手間がかかる

そこで、本研究では以上の三つの問題点の解決を目指し、「音のラフスケッチ」を制作する手法を提案する。

3. 提案手法

3.1 概要

提案手法は、2.1 で述べた4種のパラメータの指定を以下のように行う。

- (1) ユーザが求める音色をオノマトペで表現し、システム画面の描き込み可能な領域（以降「キャンバス」と呼ぶ）に直接描き込むことで、音色を指定する。
- (2) キャンバスにオノマトペを描き込む位置により、発音タイミング（横軸方向）と定位（縦軸方向）を指定する。
- (3) オノマトペを描き込む大きさを音量を指定する
- (4) (1)~(3)の三つの情報をもとに、リズムトラックを再生し確認する

2.2で述べた問題点に対して、(1)により問題点(a)を、(2)により問題点(b)を、(3)により問題点(c)を解決する。以降に上記のパラメータ指定の詳細を述べる。

3.2 音色の指定

3.2.1 オノマトペの利用

提案手法では音色をオノマトペで表現して、本手法で使える音色のライブラリの中から特定の音色を指定する方法をとる。ユーザがあるリズムパターンを思いつき、そのリズムパターンで使いたい音を入力する際、ユーザが使いたい音色を表すオノマトペを、システムの画面に表示される小節や拍に対応した線が描かれているキャンバス上に描き、円で囲むことによって音色の指定をする。

イメージする音をオノマトペで表現することは、2.2の(1)で述べた声によって音を表現する方法に近い。人間がなにかイメージする音を表出しようとした場合、実際の音に似せたオノマトペを使うことが多い。このオノマトペはユーザがイメージする音の特徴を捉えており、かつ文字を用いて記録することができる。ゆえに、ユーザがイメージするリズムパタンの表出と記録のためにオノマトペを用いることで、声によって直接に表出されるイメージを、2.2の(2)で述べた手描きによる方法で手早く記録できると考えられる。

リズムトラック制作時にユーザが求める音を表現するときに、ユーザが求める音が電子音や環境音だった場合、「ピッ」という音や「シャッ」という音などのようにオノマトペで直接表現することがある。このようなオノマトペはユーザが求める音のイメージと直接繋がっているため、「digitalsound01」などと名前のついたサウンドファイルをライブラリから探すよりも、オノマトペを描き込むことで直接音色を指定する方が音色を探す際の手間が減り、2.2で挙げた問題点(a)が解決できると考えられる。

なお、オノマトペという語には、擬音語と擬態語の両方の意味が含まれる。しかし、擬態語はリズムパターンに使う音を表現するものとしてはほぼ使われない。そこで本稿では、以降オノマトペを擬音語と同じ意味で使うこととする。

3.2.2 オノマトペと音色の対応

提案手法はリズムパタンの表出が目的であり、長時間鳴り続ける持続音は必要ないため、オノマトペと対応する音

色は比較的短時間（～1000ms程度）で減衰する音とする。このような短時間で減衰する音のうち、音楽素材として使われそうなものを使用できる音色とし、その音色に対応するオノマトペを本手法で使用できるオノマトペとする。

短時間に減衰する音とそのオノマトペによる表現については、衝突や摩擦に起因する音に関する比屋根らによる研究[5]がある。比屋根らはこの研究の中でそのような音を「単発音」と呼んでおり、それらの音にどのようなオノマトペが一般的に使われるかを述べている。このように、日本人の間ではオノマトペと音の対応にある程度の一般性がある。その一般性をリズムトラックを制作する際の音色の手早い指定に利用できると考えられる。

また、本手法では似たような音でも音程が違う音を指定するときにオノマトペを変化させることで音色を指定することを可能とする。田中らによる研究[6]では、ある音をオノマトペで表現する際の第1音の母音（「ピ」と「ポ」と表現される音を例にとると、/p/に続く/i/と/o/）の変化は、その音の支配的な周波数に近いフォルマント周波数を持つ母音が選択された結果である、と報告している。この母音の変化とフォルマント周波数の変化の関係をみると、第2フォルマント周波数が低い方から順に/o/→/u/→/a/→/e/→/i/となっている。すなわち、「ピ」と表現される音は「ポ」と表現される音と比べてその支配的周波数が高くなっていると推測される。そこで、本手法においても、ユーザが求める音をオノマトペで指定する際、似たような音で子音が共通しているオノマトペについて、そのオノマトペで使われている母音が/o/→/u/→/a/→/e/→/i/と変化するに従って、指定される音色の音程を低いものから高いものに変化させる。例えば、ユーザが「ここで使う音は『ピ』というよりも『ポ』という感じ」などと考えた際に、これらの似たような音色で音程が違う音を、それぞれ母音が異なるオノマトペで表現することで、ライブラリの中から選択する。これにより、ユーザはより手早く自分の求める音色を指定できると考えられる。

3.3 発音タイミングの指定

発音タイミングの指定は、3.2で述べたように、キャンバスにオノマトペで音色を指定し描き込んだときに同時になされる。これにより、ユーザがイメージする音の発音タイミングを指定するのにかかる手間が減り、2.2で挙げた問題点(b)が解決できると考えられる。キャンバスの横軸が時間軸に対応し、キャンバスの右側にいくほど発音タイミングは遅くなる。

3.4 音量・定位の指定

オノマトペで表現された音の音量は、キャンバスに描き込むオノマトペとそれを囲む円の大きさを指定する。また、キャンバスの縦軸を音の定位の左右に対応させ、オノ

マトペを描き込む位置の上下によって定位の指定を行う。

提案手法のように、オノマトペの位置や大きさを直接変化させて音量・定位のパラメータを指定することで、一般的な DAW 環境において音量・定位の指定時に必要となる手間を省き、ユーザがイメージするリズムパターンで使われている個々の音の音量・定位などの情報を表出しやすくなると考えられる。これにより、ユーザのイメージに近いリズムトラックを手早く制作することができ、2.2 で挙げた既存の DAW 環境での問題点 (c) が解決できる。

3.5 リズムトラックの再生

制作したリズムトラックを聞き返すことで、ユーザは自分のイメージするリズムパターンを確認する。そのためには記録されたリズムパターンが再生できる必要がある。

提案手法ではオノマトペをキャンバスに描き込んだ後、再生ボタンを押すとそれらの情報をもとにすぐさまリズムトラックとして再生される。これにより、ユーザが制作したリズムトラックがイメージ通りのものであるかすぐに確認でき、それに合わせてギターなどを演奏する、ということが容易になる。再生時には、一般的な DAW と同じような再生ヘッドが 1 小節目の先頭から動き始め、キャンバス上に配置されたオノマトペの左端に到達した時点でそのオノマトペに対応した音を鳴らす。

4. Onomatrack

3. で述べた提案手法に基づき、音色をオノマトペで表したオブジェクトを画面上に配置してリズム入力を行うシステム “Onomatrack” のプロトタイプを、Apple 社の iPad で動作するアプリケーションとして実装した。

なお、このプロトタイプでは、リズムトラックを制作する際のオノマトペの使用が、ユーザがイメージするリズムパターンの表出に対してどの程度有効かをまず調べるために、3.2.1 で述べたオノマトペを手描きでシステムの画面上のキャンバスに描き込む方法ではなく、パレットからオノマトペを選択し、それをドラッグ&ドロップでキャンバス上に配置することで発音タイミングと音色を同時に指定するシステムとした。キャンバス上に配置できる音色を表すオノマトペが描かれた画像を、以降「オノマトペオブジェクト」と呼ぶ。

使用できる音色の数は 40 個とした。以下にこれらの選出方法を説明する。

- (1) 50 音順に 3 モーラ*2 までのオノマトペを考え、そのうち音楽素材に使いそうなものを残した。例外は 4 モーラの「カラカラ」のみである
- (2) それぞれのオノマトペが表す音を既存の音楽素材から選択、または制作した。これらの音色は一般的な

ラムセットに含まれる音の他、パーカッション音、電子音、環境音で、再生時間が 1000ms 程度のものとした。その結果、一つのオノマトペに対して複数の音楽素材が対応する場合も含め、約 120 個の音楽素材を用意した

- (3) この中から筆者が一般的にリズムパターンを制作する際に必要だと判断したもの 40 個に絞った

なお、これらの本システムで使用できる音楽素材は、

- Apple 社の GarageBand のループ素材、または音源を筆者が編集、加工したもの
- インターネット上のフリー効果音配布サイト [7][8] からダウンロードしたもの、またその一部を筆者が加工したもの
- 筆者が DAW 環境で制作したもの

である。

また、本システムで使用できる音色のうち、いくつかの音色として、似た音だが音程が違う音色を、基準となる音のピッチを変更して制作した。これは、3.2.2 で述べたように、ユーザが「ここで使いたい音色は『ピ』というよりも『ポ』という感じだな」などと考えて音色を選ぶ場合があると考えたためである。

音量の調整はオノマトペオブジェクトの大きさによって 10 段階で調整できるようにし、オノマトペオブジェクトの上下に配置したハンドルを操作することで行う。また、定位の調整はキャンバスを分割する領域のどこにオノマトペオブジェクトを置くかで行う。音が定位する位置はオノマトペオブジェクトが置かれた領域ごとに上から順に以下のようにした。

- R90° : 完全に右に定位する (イヤフォンで音を聴いた場合、右側からしか音がしない)
- R45° : 右斜め 45° に定位する
- CENTER : 中央に定位する (イヤフォンで音を聴いた場合、音が左右で同じ大きさに聴こえる)
- L45° : 左斜め 45° に定位する
- L90° : 完全に左に定位する (イヤフォンで音を聴いた場合、左側からしか音がしない)

前述のように、音量の分解能を 10、定位の分解能を 5、とした理由は、思いついたリズムパターンを記録した「音のラフスケッチ」を制作する際に、必要以上に細かい調整ができる必要はないと考えたからである。それらの細かい調整に時間をかけるよりも、思いついたイメージをあとから確認できる程度の分解能で、手早く入力ができることを優先した。

実装したシステムの操作画面の一例を図 1 に示す。

5. 評価実験

5.1 実験目的

4. で述べた “Onomatrack” のプロトタイプを使用した場

*2 音韻論上一定の時間的長さを持った音の文節単位。

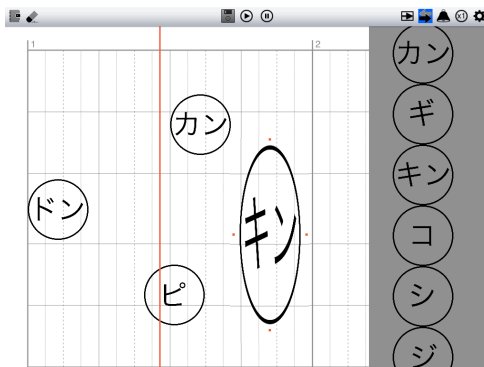


図 1 Onomatrack の画面

表 1 被験者の属性

被験者 (楽器)	年代 (社会的立場)	複雑なリズムパターンを 記録したいと思った経験
A (ドラム)	20代 (社会人)	ある
B (ギター)	20代 (学生)	ある
C (ベース)	30代 (社会人)	ない
D (ギター)	20代 (学生)	ない

合に、一般的な DAW 環境を使用した場合と比較して、

- イメージしたリズムパターンが手早く表現できたか
- 制作したリズムトラックにユーザが納得できたかを調べる。

5.2 手順

プロトタイプを被験者に1週間使用させ、その後インタビューを行った。このとき、被験者にはプロトタイプを用いて評価実験期間中に最低三つのリズムトラックを制作する、というタスクを課した。その他はできるだけ毎日システムを使用すること以外に特に制限は設けなかった。

一般的な DAW 環境での音楽制作の経験があり、バンドでの演奏経験がある4名を被験者とした。表1に4名の被験者の属性をまとめる。

5.3 結果と考察

「イメージしたリズムパターンが手早く表現できたか」という点について、被験者 A, B は手早く表現できたと答え、C, D は手早くは表現できなかったと回答した。このことから今回実装したプロトタイプは、今までに複雑なリズムパターンのイメージを記録しておきたい、と思ったことのある A, B からは、既存のリズムトラック制作環境と比較し

てイメージしたリズムパターンを手早く表現できると評価されたことがわかる。A, B はともに、ドラッグ&ドロップという入力自体によって既存の環境と比較して手早く入力できたと回答した。また、B 以外の被験者は「ドラッグ&ドロップによるオノマトペの入力よりも、手描きによるオノマトペの入力の方がイメージするリズムトラックを手早く作れるのではないかと回答している。B は手描きによるオノマトペの入力について「ドラッグ&ドロップによる入力はそのまま、さらに機能を追加する場合、手描きによる機能をオプションとしてつける方がよいのではないかと述べた。

イメージするリズムパターン通りのリズムトラックを手早く制作できたか否かに関わらず、全ての被験者が既存のリズムパターン制作環境と比較して音色の選択が手早くできたと答えていることから、提案手法の音色の指定にオノマトペを使用する方法は既存のリズムパターン制作環境よりも手早く音色を選択できることがわかった。

プロトタイプのオノマトペと再生された音の対応は納得できるものだったか、という質問に対して A, C が「納得できた」と回答し、B, D が「一部疑問に思うものもあったが、だいたい納得できた」と回答した。また、「ダン」と「ドン」など似たようなイメージの音を想像させるオノマトペについて「どちらが自分が求めている音かわからなくなることがあった」と回答した被験者もいた。これらのことから、オノマトペと再生される音の対応は、ある程度共通した認識をもっているものの、個人差があり、イメージする音を手早く指定するためには、ユーザごとにそれらの対応をカスタマイズできる必要があると考えられる。

また、リズムイメージを直接表出することを目指したシステムとして、

- オノマトペオブジェクトどうしが重なってしまったときに視認性や操作性が悪くなる

という問題点を C が指摘しており、これは解決しなければならない課題である。

さらに、今回実装したプロトタイプに対する不満点として、

- 範囲選択によるオノマトペオブジェクトの一斉操作ができないこと
- アンドウ・リドゥ機能がないこと
- コピー&ペースト機能が使いづらかったこと
- 選択されているオノマトペオブジェクトがわかりづらいこと

などの「音楽編集ソフト」としての機能不足が全ての被験者に共通していた。今回実装したプロトタイプでイメージするリズムパターンを手早く表現できなかったと回答した C, D は、これらの機能不足に対する不満がシステムの評価に繋がっていることも指摘している。これらの機能も、イメージするリズムパターンを手早く表出する際に手早く作

業を行うために必要な機能であると考えられるので、この点の改善も必要である。

その他、全ての被験者が、今回実装したプロトタイプをまた使いたいか、という質問に対して「また使いたい」と答えている。その理由として、イメージしたリズムパターンを手早く表現できるという意見があった他に多かった意見として、

- 特殊な音が多く含まれており、面白いリズムトラックが制作できる
- パレットに並んでいる音からアイデアを膨らませてリズムトラックを制作することができる

などの「発想の補助としてのシステム」としての側面を評価するものがあり、この観点でも一定の有用性はあるものと考えられる。

6. まとめ

本稿では、音楽制作経験のあるユーザを対象に、ユーザが思いついた複雑なリズムパタンのイメージを活かした音楽制作を支援するために、オノマトペを用いてリズムパタンのイメージを手早く表出し、記録することを可能にする手法を提案した。また、提案手法の有用性を示すために、提案手法に基づくシステム“Onomatrack”のプロトタイプを実装し、それを用いた評価実験を行った。その結果、既存のDAWを用いる場合と比較して、思いついた複雑なリズムパタンのイメージを記録したいと考えているユーザから、実装したプロトタイプを使用することで、イメージするリズムパターンを手早く表出し記録することができるという評価が得られた。また、リズムパタンのイメージを手早く表出することができたか否かに関わらず、既存のDAW環境と比較して、オノマトペを用いて音色の指定が手早くできるという評価が得られた。

今後の課題としては、手描きによるオノマトペの入力の実現が挙げられる。実装したプロトタイプでリズムトラックの制作が手早くできなかったと回答した被験者も含め、複数の被験者が「手描きによるオノマトペの入力の方がイメージするリズムトラックを手早く作れるのではないか」と回答していた。このことから、オノマトペをキャンバスに直接描き込むという方式を実装した上での再評価が必要である。

また、オノマトペで音色を指定する際、そのオノマトペから想像される音に個人差があるため、これらの対応をユーザがカスタマイズできるようにすることや、オノマトペと音の対応についてさらなる調査が必要である。その他、オノマトペオブジェクトがキャンバス上で重なり視認性が悪くなることや、基本的な編集機能の不備が評価に影響を与えていることから、これらの改善も必要である。

参考文献

- [1] 中野倫靖, 緒方淳, 後藤真孝, 平賀謙: ロドラム認識手法とそのドラム譜入力システムへの応用 (音楽情報), 情報処理学会論文誌, vol.48, no.1, pp.386-397, Jan. (2007).
- [2] Apple: GarageBand, (online), 入手先 (<http://www.apple.com/jp/ilife/garageband/>)(2013.07.12).
- [3] Farbood, M., Kaufman, H. and Jennings, K.: *Hyper-score: A graphical sketchpad for novice composers*, IEEE Computer Graphics and Applications, vol.24, no.1, pp.50-54, Jan. (2004).
- [4] Forsberg, A., Dieterich, M. and Zeleznik, R.: *The music notepad*, Proceedings of the 11th annual ACM symposium on User interface software and technology (UIST '98), pp.203-210, Nov. (1998).
- [5] 比屋根一雄, 澤部直太, 飯尾淳: 単発音のスペクトル構造とその擬音語表現に関する検討, 電子情報通信学会技術研究報告. vol.97, no.586, pp.65-72, March (1998).
- [6] 田中基八郎, 松原謙一郎, 佐藤太一: 異音の表現における擬音語の検討: 衝突音等の単発音やうなり音の場合, 日本機械学会論文集. C 編, vol.61, no.592, pp.4730-4735, Dec. (1995).
- [7] 小森平: フリー効果音初めての方へ (利用規約), (online), 入手先 (<http://taira-komori.jpn.org/welcome.html>) (2013.07.12).
- [8] 笠畑栄樹: On-Jin 利用規約, (online), 入手先 (<http://www.yen-soft.com/ssse/kiyaku.php/>) (2013.07.12).