Vol.2022-MUS-134 No.55
Vol.2022-SLP-142 No.55
2022/6/18

Transformerを用いたオノマトペ音声からの爆発音合成の試み

滝沢 力[1]　平井 重行[2]

概要：アニメや映画, ゲームなどの制作現場では, サウンドエンジニア・クリエイターが, 経験や知識・技能により効果音を選定・収集・生成・編集している. 最近は, プロ以外の人による作品制作は盛んに行われるが, 効果音の選定や編集による表現は素人には容易ではない. ただ, オノマトペ（擬音語）として音声で音のニュアンスも含めた効果音を表現することはある程度可能である. そこで, 本研究では, オノマトペ音声を用いた効果音合成手法の確立を目指す. 特に, 様々な種類やニュアンスの表現が含まれる爆発音に焦点を当て, その音響合成手法について取り組む. ここでは, 映画やアニメーション等で利用される爆発音の音響データ多数と, それらを口頭でオノマトペとして発話した音声データ多数を用意した. そして, 系列変換モデルである Transformer でメルスペクトログラム画像を学習し, 爆発音合成（音声から効果音への変換）を試みた. 本稿では, Transformer での学習およびメルスペクトログラムからの音響合成モデルの学習について述べ, 現状で得られている生成結果について報告する.

キーワード：オノマトペ, メルスペクトログラム, 音響合成, Transformer, 効果音

1. はじめに

映画やアニメ, ゲームなどでは, 場面に応じて様々な効果音が使用されている. それらの音響制作現場では, 既存の効果音データベースの音素材を選択したり, 必要な効果音そのものや編集・加工を前提とした音素材を収録することが行われる. そして, 場合に応じて選択・収集した効果音に対し, 波形編集や周波数成分の調整, 残響効果などのエフェクト処理なども含め, データを加工・編集して, 作品で利用される効果音へと仕上げていく [1][2][3]. これら音響制作の技術者は, 効果音データベースの音の種類・バリエーションについて熟知していたり, それを検索・選択するための知識やスキルを持っていたり, 収録・編集技術についても同様に知識やスキルを持っている. しかし, 経験の浅い人がこのような作業を通して効果音を選定・制作するには膨大な時間や労力が必要になる. しかし, オノマトペ（擬音語）として音声で音のニュアンスも含めた効果音を表現することは, 多くの人にとって容易である. そこで, 本研究では, オノマトペの発話表現を用いて効果音合成を行い, 生成音のニュアンス制御を可能にする手法の確立を目指す.

2. 関連研究

岡本らによる Onoma-to-Wave[4] では, テキスト音声合成技術のように, オノマトペのテキストからの環境音合成を試みた. ここでは生成モデルとして, 系列変換モデル（エンコーダ・デコーダモデル）を使用している. エンコーダにオノマトペテキストを入力し, 一層の Bidirectional LSTM により音素ごとの特徴ベクトルを得る. 二層の LSTM で構成されたデコーダでは, エンコーダで得られた特徴ベクトルから音響特徴量を推定していく. また, エンコーダの出力に対して, 環境音の種類を示す音響イベントラベルを付け加えることで, 音の種類を表現できるようにしている. これらの処理により, オノマトペテキストからバリエーションを考慮した音響合成を実現している. ただ, この生成モデルには再帰構造を利用しており, 長い系列の特徴をうまく捉えられないという課題があった. 一方, 系列変換モデルである Transformer[5] は Attention 機構により長期的な特徴を捉えることが可能であり, 系列長が長い場合でも機械翻訳 [5] や音声合成 [6] などで高い変換性能を示している. そこで, 岡本らは Transformer を用いたオノマトペからの環境音合成を行うことでこの問題の改善を試みている [7]. モデルの学習では, オノマトペテキストの音素系列を三層の CNN で構成された Encoder Pre-net に入力した結果を Transformer のエンコーダに入力し, 処理していく. デコーダ側では, 環境音のメルスペクトログラムを二層の全結合層で構成された Decoder Pre-net に入力し, Encoder Pre-net と同等の次元空間に射影する. そのデータをデコーダに入力し, エンコーダ側の音素との対応関係を Transformer と

[1] 京都産業大学大学院 先端情報学研究科
[2] 京都産業大学 情報理工学部

して学習することで、メルスペクトログラムを推定する．この手法により，長期的系列でも依存関係を考慮した合成を実現している．

3. 提案手法

3.1 概要

本手法では, Transformer-TTS[6] や関連研究の後者 [7] などの音声・音響合成で高い変換性能が示されている Transformer[5] を変換モデルとして使用する．先述したように, 関連研究 [4],[7] は, テキスト（オノマトペの音素列）から環境音への変換を行っていたため, テキスト音声合成技術による環境音合成であるとみなすことができる．また, Onoma-to-Wave[4] では音響イベントラベルとしてテキスト以外の情報を付与することで, 合成音のバリエーションを考慮していたが, 合成音のニュアンスを示す韻律情報は両研究 [4],[7] 共に用いていない．そこで, 本手法では, 多くの人が, 聞こえた音の韻律情報（音のニュアンス）を発話表現することが容易であることからオノマトペの発話音声を扱うこととした．実際にモデルに入力するデータを音声・効果音データそれぞれのメルスペクトログラムの画像とすることで, 音素列などのアノテーションを必要とせず画像から画像への変換を行うことができると考えられる．

今回は, 数ある効果音の中から, 様々な種類やニュアンスの表現が含まれる爆発音に焦点を当て, 合成を試みた．

3.2 データセット

本研究で用いるデータセットについて説明する．まず, 爆発音素材としては, インターネット上で公開されている効果音データ [8]〜[17] や, 販売されている効果音集[*1]から, 734 個のオーディオファイルを用意した．これらに対し, 爆発音を一つずつ聞きながら, 音素列としては記述が困難な細かな発音ニュアンスや韻律も含めたオノマトペ（擬音語）音声を録音した．録音は爆発音毎に 2 回ずつ行い, オノマトペ音声データとしては, 734 種類 × 2 回で計 1468 個用意した．これら, 爆発音およびオノマトペ音声すべてをサンプリング周波数 44.1kHz, 量子化ビット数 16bit のオーディオファイルとして学習に用いた．**図 1** に, 爆発音および対応して発話したオノマトペ音声のそれぞれのメルスペクトログラムを示す．

3.3 提案モデル

モデル全体の処理概要を**図 2** に示す．

図 2 は大きく前処理, Transformer, 波形合成の 3 つの処理に分けられる．前処理では, 音声・爆発音データを図 1 のようなメルスペクトログラム（二次元画像）に変換する．メ

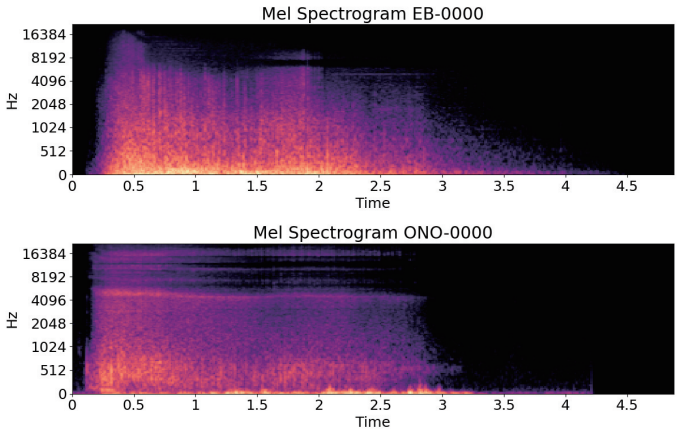


図 1: データセットのメルスペクトログラム（上が爆発音, 下がオノマトペ音声）

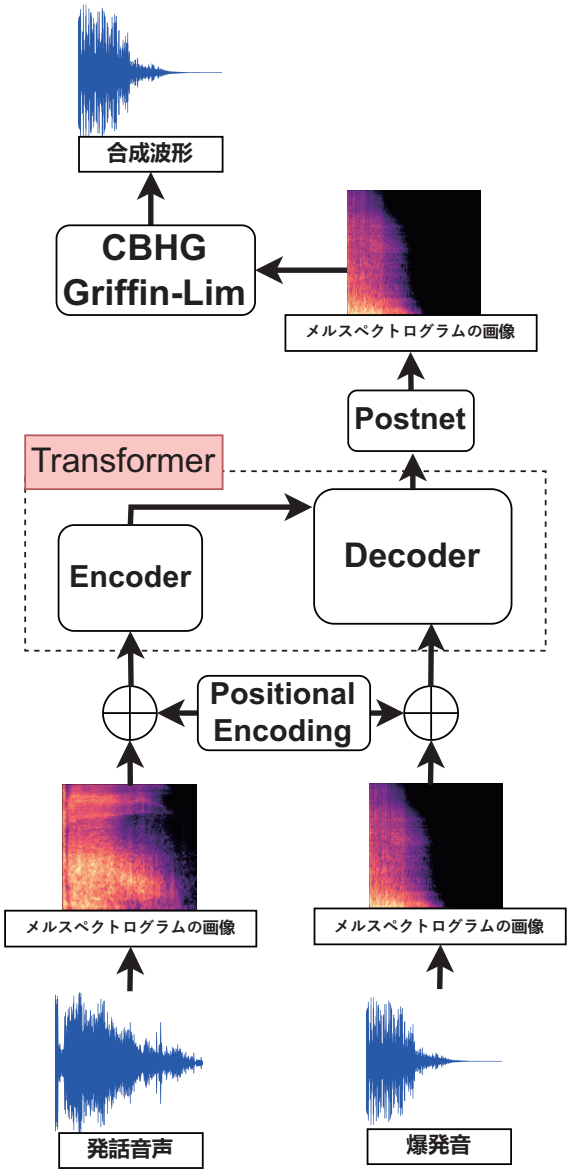


図 2: 提案モデルの学習・生成時の処理フロー

[*1] SONICWIRE ”爆発・災害に関するサウンドを中心に収めた効果音パック” ＜ https://sonicwire.com/product/A9582 ＞ （最終アクセス日：2022 年 1 月 5 日）

ルスペクトログラムは, それぞれ Transformer のエンコーダ・デコーダで処理され, デコーダから予測されたメルスペクトログラムの残差を Postnet にて計算していき, 音声画像から爆発音画像への変換を学習する. 予測されたメルスペクトログラムは CBHG モデル [18] により振幅スペクトログラムに変換され, Griffin-Lim アルゴリズム [19] に基づき波形合成される.

4. モデルの学習・生成

4.1 学習

モデルの学習では, オノマトペ音声から爆発音を予測する Transformer と, メルスペクトログラムから振幅スペクトログラムに変換する CBHG モデルの二つを学習する必要がある. Transformer の学習には, 先述した音声・爆発音データのメルスペクトログラムを用いる一方で, CBHG ではメルスペクトログラムと振幅スペクトログラムを用いて学習を行う.

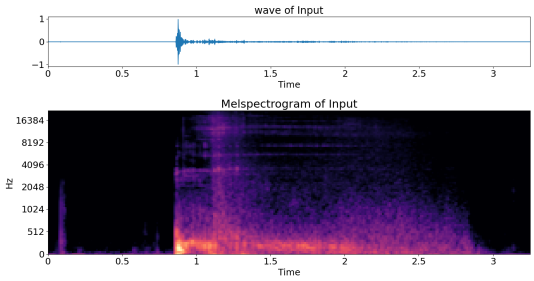
今回の学習では, それぞれのモデルをエポック数 2000, バッチサイズ 16 とし, データセッすべてを用いて学習を行った. 計算機環境としては Nvidia Tesla P100[*2]の GPU を用い, 深層学習のフレームワークには PyTorch を用いた.

4.2 生成

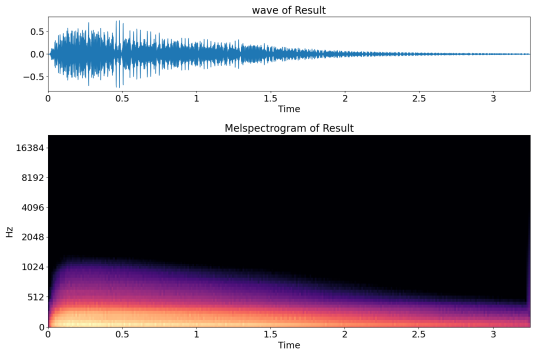
学習済みモデルに未知のオノマトペ音声を入力し, それに応じた爆発音を予測させていく. ここでは, ニュアンス制御が可能かを確認するため, いくつかの音韻の違う音と共に, 同じ音韻でも発話長など韻律が違うものも入力音声として 11 種類用意し, 結果を確認した. **図 3** は, 遠くでなっている様な爆発音を模したオノマトペ音声と, 生成結果を示している. **図 4** は, 複数回爆発している様なオノマトペ音声と, 生成結果を示している. **図 5**〜**図 7** では, それぞれ「どかーん」・「ばーん」・「どーん」の様に音韻が異なる 3 種類を用意し, それぞれに対して、大中小で音の継続時間を変化させた場合で比較している.

5. 考察

図 3 は，遠くで爆発音が鳴っている様な入力音声とその生成結果である．この図では，入力音声の開始時点でのアタック部分のスペクトルがあるのに対し、出力結果ではそのようなアタック部分がないが、低域の周波数成分が強調されたものとなっている．ただ、音として聞くと，遠くで爆発している様な音が合成されており，アタックがなくともニュアンスとして表現できている部分があると言える．

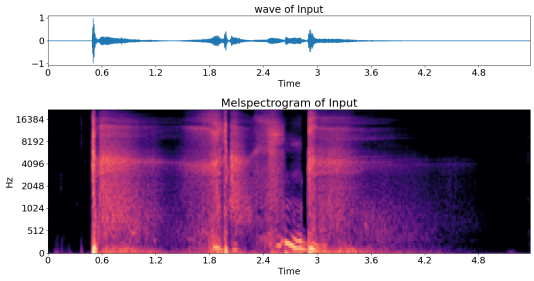


(a) 入力音声

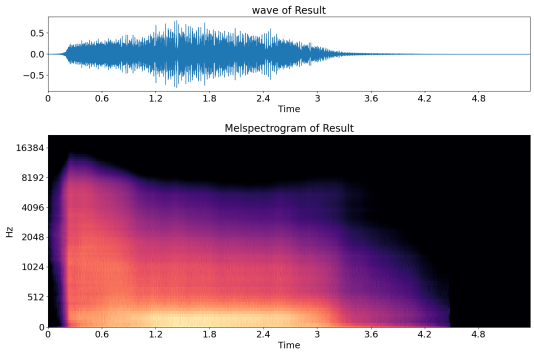


(b) 出力結果

図 3: 遠くでなっている爆発のような音声からの合成



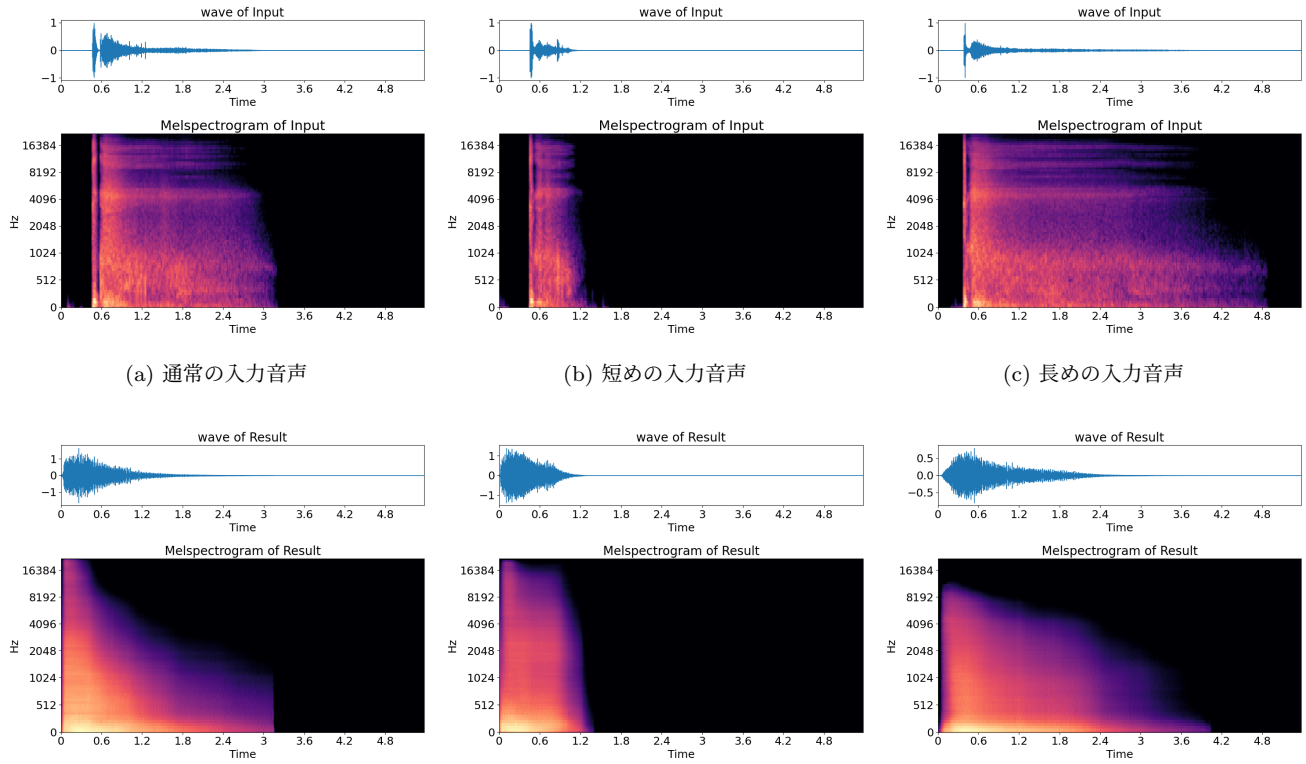
(a) 入力音声



(b) 出力結果

図 4: 複数回爆発しているような音声からの合成

[*2] https://www.nvidia.com/ja-jp/data-center/tesla-p100/



(a) 通常の入力音声　(b) 短めの入力音声　(c) 長めの入力音声

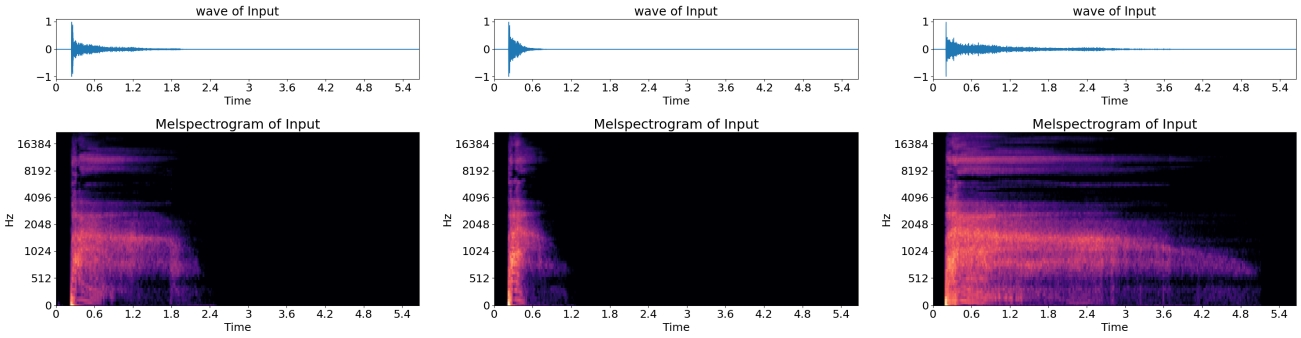
(d) 通常の生成結果　(e) 短めの生成結果　(f) 長めの生成結果

図 5: 「どかーん」の音声と生成結果

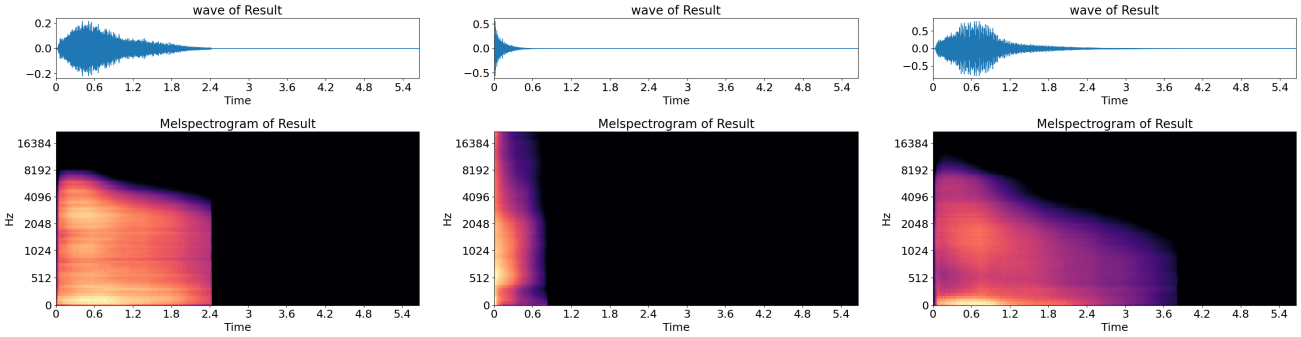
次に，図 4 は爆発音が時間差で複数回鳴るような入力音声を用いた際の生成結果である．図 4(b) の出力結果からは，最初の爆発音の鳴り出しのアタックは合成できているものの，途中で鳴る爆発音のアタックがない結果となってしまった．続いて，図 5〜図 7 は，音韻が異なる爆発音のオノマトペ表現をそれぞれ入力した結果を示す．音韻としては敢えて表現するとそれぞれ「どかーん」,「ばーん」,「どーん」と記述できる．そして，これら 3 種類については、ニュアンスの違いの観点から，韻律の一つとして発話長を変えたものも用意して合成した．それぞれの時間長さに応じた爆発音が合成されていることが、これらの図から確認できる．また，これらはそれぞれの音韻の違いに相当するニュアンスの違う爆発音としても合成できている．以上の結果から，今回行ったメルスペクトログラムを用いた Transformer での学習モデルによって，オノマトペの入力音声からニュアンスの違う爆発音の合成制御が可能であると考えられる．ただ，図 4 に示す結果のように，複数のアタック音が含まれる入力音声に対しては，対応する生成音が得られていないことも確認できた．これは，学習に使用したデータセットにこのような複数の爆発音が含まれるようなデータが少ないことが理由として考えられる．今後は，そのようなデータも増やしたデータセットとして学習を試みる必要がある．また、今回のデータセットは入力用として収録した音声は男性 1 名のもののみであった．同じ爆発音でも別の人であれば別のオノマトペ音声として表現することが考えられることから，複数人での入力用音声を収録してデータを増やした形で学習を試みることも必要と考えている．加えて，モデルの表現力向上にむけて，爆発音の種類も増やす形でデータセットを増やすことも必要と考える．

6. おわりに

本研究は，効果音作成や編集などのスキルがない人でも効率的に効果音の作成が可能となることを目指し，ニュアンスの表現も含めたオノマトペ音声による効果音の合成を行う手法について提案した．ここでは，用いる手法としてメルスペクトログラム画像のみを入力とする Transformer を用いている．具体的な効果音の題材として爆発音を対象に，オノマトペ音声を収録してデータセットを構築し，学習に用いた．そして爆発音の生成結果からは，音韻の違いや韻律としての発話長の違いなどで爆発音のニュアンスが制御できることを確認した．ただ，うまく合成できていない例も確認できたため，今後はデータセットの拡張を含め，合成の精度向上と音質の向上を目指してゆきたい．

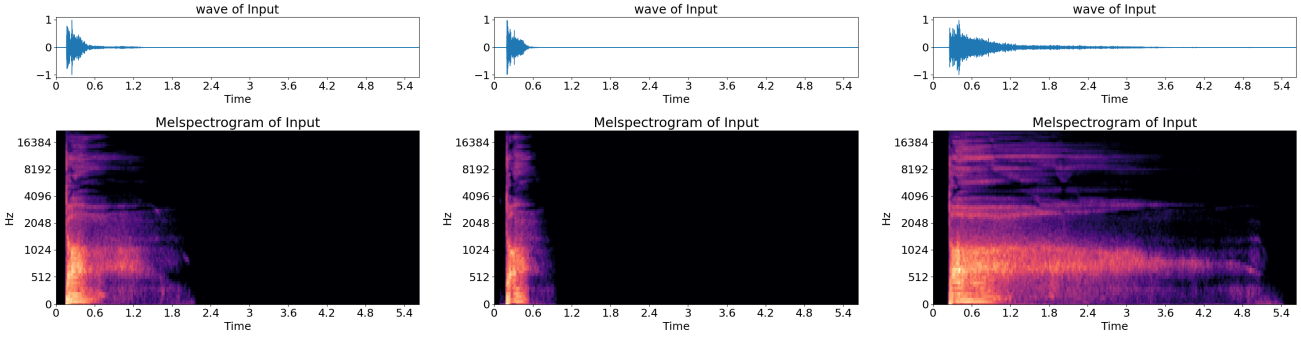


(a) 通常の入力音声　(b) 短めの入力音声　(c) 長めの入力音声

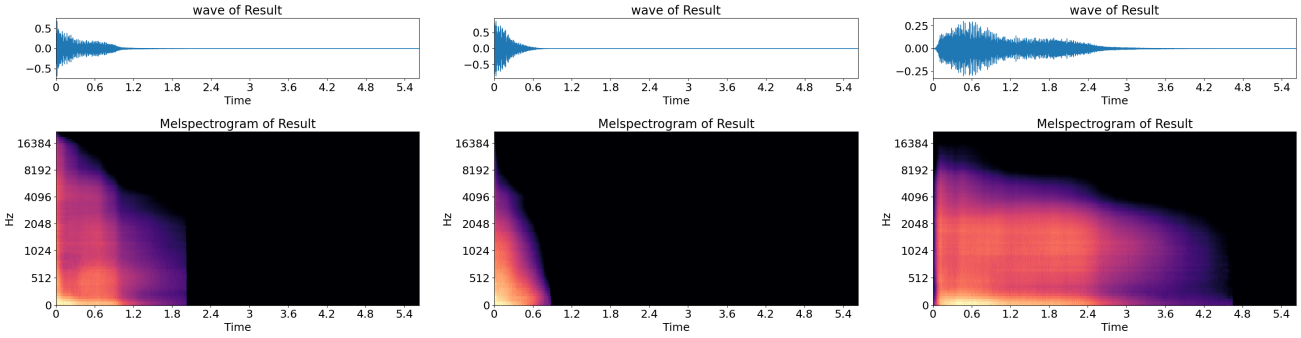


(d) 通常の生成結果　(e) 短めの生成結果　(f) 長めの生成結果

図 6: 「ばーん」の音声と生成結果



(a) 通常の入力音声　(b) 短めの入力音声　(c) 長めの入力音声



(d) 通常の生成結果　(e) 短めの生成結果　(f) 長めの生成結果

図 7: 「どかーん」の音声と生成結果

参考文献

[1] 木村哲人, ＜キムラ式＞音の作り方, 筑摩書房, 1999.
[2] 小川哲弘, サウンドエフェクトの作り方 [改訂版], 工学社, 2021.
[3] デイヴィッド・ゾンネンシャイン, Sound Design 映画を響かせる「音」のつくり方, フィルムアート社, 2015.
[4] Y. Okamoto, K. Imoto, S. Takamichi, R. Yamanishi, T. Fukumori, Y.Yamashita, "ONOMA-TO-WAVE: ENVIRONMENTAL SOUND SYNTHESIS FROM ONOMATOPOEIC WORDS," arXiv: 2102.05872v3, 20 Oct 2021
[5] A. Vaswani, et al., "Attention is all You need," Proc.NIPS, pp.6000-6010,2017.
[6] N. Li, et al., "Neural speech synthesis with transformer network," Proc. AAAI, pp.6706-6713,2019.
[7] 岡本悠希, 井本桂右, 高道慎之介, 福森隆寛, 山下洋一, "Transformer を用いたオノマトペからの環境音合成," 日本音響学会講演論文集 P.943-946, 2021 年 9 月
[8] E エフェクツ ＜ https://esffects.net ＞ （最終アクセス日：2021 年 12 月 13 日）
[9] On-Jin 音人 "" ＜ https://on-jin.com/sound ＞ （最終アクセス日：2021 年 12 月 27 日）
[10] クラゲ工匠 ＜ http://www.kurage-kosho.info ＞ （最終アクセス日：2021 年 12 月 27 日）
[11] 効果音ラボ ＜ https://soundeffect-lab.info/sound ＞ （最終アクセス日：2021 年 12 月 27 日）
[12] OtoLogic ＜ https://otologic.jp/free/se ＞ （最終アクセス日：2021 年 12 月 27 日）
[13] 効果音工房 ＜ https://umipla.com/＞ （最終アクセス日：2021 年 12 月 27 日）
[14] 魔王魂 ＜ https://maou.audio/＞ （最終アクセス日：2021 年 12 月 27 日）
[15] VSQ plus+ ＜ https://vsq.co.jp/plus/＞ （最終アクセス日：2021 年 12 月 27 日）
[16] HURT RECORD ＜ https://www.hurtrecord.com/＞ （最終アクセス日：2021 年 12 月 27 日）
[17] Sounds-mp3.com ＜ https://sounds-mp3.com ＞ （最終アクセス日：2021 年 12 月 27 日）
[18] Y. Wang, R. Skerry-Ryan, D.Stanton, Y.Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y.Xiao, Z. Chen, S. Bengio, Q. Le, Y.Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis", in Proc. Interspeech, 2017, pp. 4006-4010.
[19] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," IEEE Transactions on Acoustices, Speech and Signal Processing, pp. 236-243, 1984.