AmpLab 2: Freesound Audio Mosaicing - Qin Liu

1. Introduction

This document outlines the outcomes of experiments in audio mosaicing, leveraging the FreeSound API to source audio for reconstruction. Inspired by Robert Silvers' Photomosaics [1], where numerous semantically connected photographs are pieced together, audio mosaicing involves piecing together fragments of sound from various clips to reconstruct a target audio piece. This process is grounded in the concept of semantic similarity, merging audio features from both target and source clips to produce a version that echoes the original sound. Audio mosaicing offers limitless potential for creativity, providing users with an expansive selection of sources and targets, alongside the features to be extracted, based on the desired end result, whether it be replication or abstraction. This document delves into two specific uses of audio mosaicing: (1) investigating rhythmic elements to recreate the well-known amen break-beat and (2) conducting tonal melodic reconstruction using a piano sample, with influences from the JP-X Roland synthesizer and inspired by musicians like Aphex Twin and Boards of Canada. The structure of this report includes a methodology section detailing the approaches for rhythmic and melodic reconstruction, followed by a section presenting the outcomes, conclusions, and a discussion on the efficacy of each method implemented.

2. Methodology

The extraction of features from both the source and target audio clips is conducted through the Python interface of Essentia, a comprehensive open-source library that provides tools for analyzing, describing, and synthesizing audio.

2.1. Target and Source Audio Selection

For this project, we focused on reconstructing two distinct audio samples: a segment of the iconic amen break-beat (https://freesound.org/people/csum/sounds/194817/), a staple in drum and bass as well as jungle music genres, and a brief melodic excerpt from a piano piece (https://freesound.org/people/tkky/sounds/486472/). Both audio samples were sourced from

FreeSound. The choice to work with the amen break stemmed from an ambition to craft a rendition of a disassembled 90s rave track. Specifically, the project aimed to replicate ambient sounds one might encounter around a rave or club scene, such as deep sub-bass and intermittent high-frequency resonances from the surrounding environment responding to the sound system's output. This concept was intriguing as it envisioned the environment not merely responding to the DJ's or sound system's output but actively mimicking and reproducing the generated sounds. To approach this, source audio was chosen based on keywords like 'bass', 'sub', '303', and 'clap', while also considering environmental sounds like the shattering of glass or background chatter, though these elements were ultimately not included in the final output.

Regarding the melodic piano segment, the intention was to mimic the ethereal and somewhat dissonant tones often found in the music of Aphex Twin and Boards of Canada. Aphex Twin, for instance, frequently employs detuned pianos to craft bewildering and eerie atmospheres in tracks like 'Vordhosbn'. To achieve a similar vibe, we sourced audio from the Roland JX-8P synthesizer, acknowledging upfront the challenge in fully matching the piano melody's pitches due to the limited variety of sounds available. This creative venture was predicated on the realization that embracing the mismatch, or incoherence, between source sounds and the target melody, while still preserving the general pitch contours, could yield interesting results.

2.2. Segmentation Features – Onset Detection

Initial experiments concentrated on segmenting the target and audio clips by their beat positions with the use of the BeatTrackerDegara function in Essentia. Although these trials yielded dependable results for both types of target clips, this strategy was ultimately not adopted in the project's final version. However, the beat tracking code is included in the accompanying Python notebook for completeness.

When analyzing the amen break beat sample, it was noted that numerous audio events occurred in the intervals between beats, such as intricate snare rolls and glitches. Consequently, segmenting the audio solely on beat markers before extracting features could lead to overlooking significant aspects of the audio. Therefore, an alternative approach was employed, utilizing Essentia's onset detection functions to segment the audio and better capture these in-between events. Three onset detection algorithms were evaluated: a complex-based method, an HFC (High-Frequency Content), and a

SuperFluxExtractor [2], based method. The comparison, especially focusing on the first three seconds of onsets for the amen break, indicated that the complex-based method was most effective for capturing the relevant onsets, leading to its selection for both target and source audio segmentation in the amen break scenario. Conversely, for the piano sample, where actual onsets were fewer, the HFC onset detection algorithm was found to provide more accurate segmentation, reflecting a better match with the reality of the piano's audio characteristics. Comprehensive visualizations of these comparisons are available in the provided Python notebooks.

2.3. Similarity Features

We observed both two sounds contain abundant rhythmic information, so in addition to the MFCCs provided by the original function – 'chose_frame_from_source_collection()', so we incorporated attack_time feature. Besides, for the piano sound that is tonal and contains much melodic content, so we included the pitch feature extracted by 'PitchMelodia()' and 'PitchContourSegmentation()' from essentia fro the piano sound. The details are coded as below.

Python

```python
def analyze_sound_for_onset(audio_path, onsets=None, audio_id=None, melodic=False):
    analysis_output = []  # Here we'll store the analysis results for each chunk (frame) of the audio file
    # Load audio file
    loader = estd.MonoLoader(filename=audio_path)
    audio = loader()

    onset_i = np.arange(0, len(onsets)-1)
    for i in onset_i:
......
        # (4) if we are considering a melodic, piece, also compute pitch
        if melodic:
            #print('melodic true')
            pitch_algo = PitchMelodia()
            pitchcontorsep_algo = PitchContourSegmentation()
            pitch_contor, pitch_confidence = pitch_algo(frame)
            onset, duration, pitch = pitchcontorsep_algo(pitch_contor, frame)
            if len(pitch) >= 1: pitch = pitch[0]
            else: pitch = 0
            frame_output.update({'pitch': pitch})

    return analysis_output

def chose_frame_from_source_collection(target_frame, df_source_frames, melodic):
     n_neighbours_to_find = 20

    if melodic: similarity_features = ['mfcc_0', 'mfcc_1', 'mfcc_2', 'mfcc_3',
                                       'mfcc_4', 'mfcc_5', 'mfcc_6', 'mfcc_7', 'mfcc_8',
                                       'mfcc_9', 'mfcc_10', 'mfcc_11', 'mfcc_12',
                                       'attack_time', 'pitch']
    else: similarity_features = ['mfcc_0', 'mfcc_1', 'mfcc_2', 'mfcc_3',
                                       'mfcc_4', 'mfcc_5', 'mfcc_6', 'mfcc_7', 'mfcc_8',
                                       'mfcc_9', 'mfcc_10', 'mfcc_11', 'mfcc_12',
                                       'attack_time']

    # Find the 10 most similar frames to the target_frame from df_source_framesdf_source_units
```

```
        query_frame = target_frame[similarity_features].values
        similar_frames = find_similar_frames(query_frame, df_source_frames, n_
    neighbours_to_find, similarity_features)

        # Choose the first one as is the most similar
        most_similar_frame = similar_frames[0]

        return most_similar_frame
```

We also did trial and error and tried featuers – loudness, zero-crossing rate, spetral centroid, etc., but the feature combinations hoding the best audio reconstruction perforamance are still [MFCC, attack time, pitch] for the piano sound, and [MFCC, attack time] for the amen break-beat sound.

3. Results and Reflections

The reconstruction of the amen break-beat successfully maintained its rhythmic quality, demonstrating the effectiveness of segmentation via onset detection. The project creatively succeeded, abstractly mimicking a club environment's acoustic properties. Future projects are encouraged to explore environmental sound modeling further using field recordings.

The piano segment's reconstruction achieved an atmospheric quality reminiscent of Boards of Canada but was limited by the randomness in pitch of some source samples. Future efforts should incorporate pitch information more selectively and consider additional filtering criteria for a more coherent reconstruction.

4. Conclusion

This exploration into audio mosaicing using the FreeSound API presents a successful application of creative sound reconstruction, achieving distinct atmospheres through both rhythmic and tonal experimentation. While some limitations were encountered, particularly in pitch coherence for the piano reconstruction, the project lays a foundation for future exploration in the field of audio mosaicing, suggesting pathways for more intricate and faithful audio reconstruction, namely, audio mosaicing.

References

[1] Silvers, R., 1996. Photomosaics: putting pictures in their place (Doctoral dissertation, Massachusetts Institute of Technology).

[2] Böck, S. and Widmer, G., Maximum Filter Vibrato Suppression for Onset Detection, Proceedings of the 16th International Conference on Digital Audio Effects (DAFx–13), 2013