BARNYARD KARAOKE: RECREATING MELODIES WITH ANIMAL SOUNDS

Patricio Ovalle
Universitat Pompeu Fabra
Barcelona, Spain
patricio.ovalle01@estudiant.upf.edu

1 Introduction

Audio mosaicing refers to the technique of reconstructing a target audio using only material from a collection of source audio. This project explores using audio mosaicing as a means to orchestrate new arrangements of popular melodies using animal sounds. The goal is to produce audio that evokes a medley of barn animals howling in the yard for a carefree evening of karaoke.

The idea takes inspiration from *Meowify*[1], a project that replaces singing voice with cats meowing. Their method consists of two steps: first, a source separation model is used to isolate the vocals from the rest of the mix; then, a timbre transfer model is used to morph the singing voice into a cat's meow before recombining the modified vocals with the rest of the mix. This project attempts to achieve a similar result via the technique of audio mosaicing. The code for this project is openly available online[2].

2 Methodology

Audio mosaicing consists of two distinct components: a target audio and a set of source audio. To decide how the source audio should be split and recombined to reconstruct the target, several key decisions must be made:

- the characteristics to emulate in a target sound
- the selection of a target sound
- the selection of a set of source audio
- a method for selecting source audio frames
- a method for recombining selected frames

The following sections address each of these concerns and describe the strategies used at each step.

2.1 Target characteristics

The first step to define the characteristics of the target sound we want to preserve. Our aim is to reconstruct a melody, so the pitches and pitch onset times of the melody in the target audio are the most essential features to extract.

To analyze the pitch contour of the predominant melody from the audio, we use the MELODIA algorithm introduced by Salamon and Gómez [2012] and implemented in the Essentia audio analysis toolkit [Bogdanov et al., 2013]. To subsequently determine the pitch onset times of the melody, we use the pitch contour segmentation algorithm in Essentia introduced by McNab et al. [1995]. This segmentation algorithm is especially convenient for our needs as it takes a pitch contour as input, which is the output of the MELODIA algorithm. A downside of this algorithm is that it requires a fine-tuning of several parameters in order to function well. For its "hopSize" and "minDuration" parameters

[1] https://github.com/gulnazaki/meowify
[2] https://github.com/p3zo/barnyard-karaoke

we choose relatively small values to ensure it captures the rhythmic nuance of the melodies. Specifically, we derive the minimum note duration using the tempo of the target tracks together with the shortest note values used in their melodies. For a track at 90 beats per minute (bpm), the duration in seconds for an eighth note is given by:

$$\frac{60\ seconds}{90\ bpm} * \frac{1}{2}\ beat = .33\ seconds$$

For each track we set the "minDuration" parameter to just below this value to account for some variation in tempo. The "hopSize" parameter must be given the same value used to estimate the pitch contour, 128 samples, since it is what maps contour indices back to times. We set the "rmsThreshold" parameter to be low as well because we deem over-detecting onsets to be better than under-detecting them for this application. The result of extraneous onsets is that more than one source frame will be used to reconstruct a given melodic pitch in the target. On the other hand the result of a missing onset is silence, which makes for a less exciting output. Each detected note is taken to extend from its onset to its onset plus its reported duration. Treating a note as ending at the following onset instead would fold the rest that follows it into the note itself, and would discard the final note of every sound. An example of an estimated pitch contour and its detected onsets are shown in Figures 1 and 2.

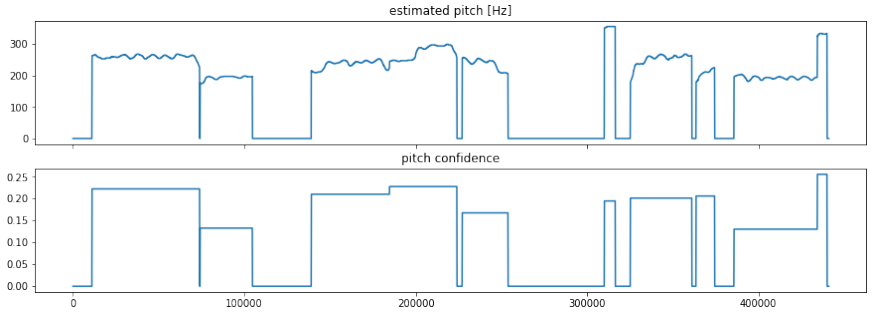

Figure 1: A pitch contour estimated using MELODIA

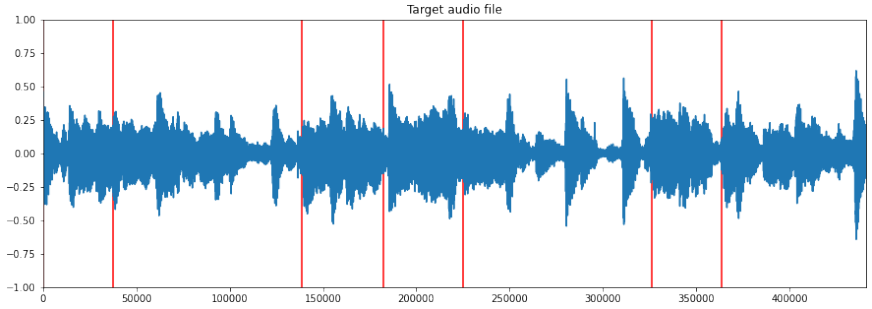

Figure 2: Original audio with estimated pitch onset time markers

2.2 Target sound selection

The next concern is to select the target sound for which we want to emulate the pitch contours. Our aim is to recreate melodies, so we choose musical tracks with well-known melodies as targets. In particular, we choose two tracks with simple vocal melodies that are easy to sing along to. The tracks are listed in Table 1 along with the timestamps used from each.

An important consideration is to select a sound with not too much complexity in the mixture to increase the effectiveness of a general pitch-tracking algorithm for melody extraction. We selected only target sounds in which the melodies were predominant and in a distinct pitch range from other instruments in the arrangement.

2.3 Source sound collection

The source collection in audio mosaicing is the set of audio from which frames can be sliced and used as material to recreate the melody of the target. Our aim is to create a set of animal sounds that are well-pitched and clearly recognizable.

Youtube ID	Title	Timestamp
V1bFr2SWP1I	Somewhere Over the Rainbow	1:05 - 1:15
_6HzoUcx3eo	Old Macdonald Had A Farm	0:15 - 0:35

Table 1: Target songs

We download our collection of audio samples from Freesound[3] using their API. The parameters used in the Freesound API queries are shown in Table 2. We select a set of animals that produce sounds in different registers overlapping with the range of the human voice. We also select animals with distinct timbres from one another in hopes of adding a bit of dynamism to the result. Each API query included a sort condition by descending user rating to get the highest-rated sounds first. In a manual exploration of downloaded results we noticed that shorter recordings were typically of higher quality than longer recordings, which often tended to be field recordings clouded with background noise and various non-animal sounds. Each collection was filtered to a short duration in order to increase the chance that the query would return single-shot sounds, and tags marking field recordings, ambiences, soundscapes and loops were excluded for the same reason. Of the frames extracted from a collection we keep only those that hold one steady pitch, measured as the standard deviation of a frame's pitch contour about its own median and required to stay within a quarter tone.

For comparison, we also create a second source collection of tuned, harmonic sounds. Specifically, we download sustained single notes from ten orchestral and keyboard instruments, filtered to a duration of 0.5 to 6 seconds and excluding sounds tagged as chords, melodies or phrases so that each recording holds a single note.

2.4 Frame selection strategy

When selecting which frames to use from the source to recreate the desired characteristics of the target audio, a similarity score is computed between each frame in the target audio and all of the frames in the source sound collection. The features are standardised to zero mean and unit variance across the source collection before the distance is computed. On raw values this distance is not meaningful: the first MFCC coefficient ranges over several hundred units while loudness ranges over thousandths, so the distance reflects little besides that one coefficient.

Candidates are then restricted to source frames that land on the target note's pitch class, in any octave, or that can be resampled onto it by at most a semitone, and among the ten closest of those we choose one at random, passing over recordings used in the last few notes, so as to add a bit more variation in the result. Frames already in tune are preferred over frames that have to be resampled, and frames nearer the melody's own octave rank first. Drawing uniformly from the ten nearest frames without that restriction does not preserve the melody: where the collection holds nothing at the target pitch, all ten can be one or two semitones away, and the reconstruction drifts out of tune.

We experimented with several sets of features for computing this similarity, namely:

- Pitch
- Pitch, loudness
- Pitch, loudness, and MFCCs

As previously mentioned, the pitch contour was estimated with the MELODIA algorithm in Essentia; the pitch of a frame is the MIDI note number reported for it by the segmentation algorithm. The loudness and MFCC features were extracted using built-in algorithms in Essentia. The goal with including loudness was to mimic the energy of the target melody, and the intent of using MFCC features was to capture the timbre of the instrument playing the melody. We computed similarity with and without MFCCs with the expectation that timbre would be less important than pitch in producing a convincing result. We also experimented with using pitch alone to see if omitting the other features would improve the general accuracy of pitch matching.

2.5 Recombination strategy

The final step in the audio mosaicing process is to define a strategy for how the selected source frames will be combined to reconstruct the target audio. We choose to recombine the selected frames at the pitch onset times determined by the segmentation algorithm. We elect to not allow the overlapping of source frames during recombination. A selected frame is frequently shorter than the target note it has to fill, and is never read past its end into whatever follows it in

[3]https://freesound.org

Count	Keyword
60	cat meow
50	dog howl
50	wolf howl
40	cow moo
40	rooster crow
30	duck quack
30	sheep baa
25	goat bleat
18	horse whinny
10	owl hoot

Table 2: Freesound queries for the animal collection, each filtered to a duration of 0.35 to 4 seconds

the source recording. The remainder is left silent, or covered by concatenating further frames of the same pitch, by looping the frame, or by time-stretching the frame to the note's length. Each placed segment is scaled to the loudness of the note it replaces and given a 5 ms fade at both ends, without which the joins between segments click audibly.

3 Results

For each reconstruction the pipeline reports the proportion of the target's duration covered by placed frames, the mean absolute difference in semitones between each target note and the source frame chosen for it, the number of frames truncated because the chosen source note was shorter than the target note, the ratio between the loudest and quietest placed segment, and how many frames were displaced by an octave or resampled into tune.

Demos of both the animal and instrument melody reconstructions can be heard at:

`https://github.com/p3zo/barnyard-karaoke/tree/main/demo`.

4 Discussion

Overall, the methodology described in this paper has a ways to go to produce the animal symphonies originally envisioned. The biggest improvement may come from assembling a well-curated sound collection. One glaring issue in the current animal sound collection is that many of the returned query results are actually sounds that are orthogonal to the query. For example, the query for "cat meow" returns results including the sound of pouring food in a cat bowl. Manual verification of the source sounds to remove non-animal sounds could go a long way. A related and larger effect is that a sound enters the collection at all only if a melodic contour can be estimated from it. Unpitched sounds yield no frames whatsoever and unsteady ones are cut by the steadiness requirement, so a substantial fraction of each downloaded collection contributes nothing to the reconstruction; the analysis step reports the sounds that yield no contour and the frames cut for unsteady pitch. Additionally, there is a high degree of variability in the audio quality of the downloaded sounds. It would be helpful to have a few attributes related to audio quality to filter on when querying Freesound. For example a "signal-to-noise ratio" attribute would help filter out sounds with a lot of background noise. A "harmonic" attribute would be helpful in determining whether a sound is suitable to be used to match any pitch.

Octave reduction recovers some of the material that would otherwise go unused. Some source frames match pitches in the target melodies but in a different octave, and labeling each source frame with its pitch class rather than its MIDI note number increases the coverage of pitch classes that can be matched exactly.

While the pitch detection algorithm used worked well on the selected target sounds, there are many complex mixtures for which it does not work well. For these cases, a source separation pre-processing step can be used to isolate the instrument playing the predominant melody, so as to simplify the input for pitch detection.

References

Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE transactions on audio, speech, and language processing*, 20(6):1759–1770, 2012.

Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, Xavier Serra, et al. Essentia: An audio analysis library for music information retrieval. In *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.* International Society for Music Information Retrieval (ISMIR), 2013.

Rodger J McNab, Lloyd A Smith, and Ian H Witten. Signal processing for melody transcription. 1995.