BARNYARD KARAOKE: RECREATING POPULAR 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 imagery of 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. See their demo entitled "Dumb by Nirvana, but Kurt Cobain's cat is singing"[2]. Their method consists of two steps: first, they use a source separation model to isolate the vocals from the rest of the mix, then they use a timbre transfer model to morph the singer's voice into a cat's meow before recombining the modified vocals with the rest of the mix.

I attempt to achieve a similar result via the technique of audio mosaicing. The code for this project is available at:

https://drive.google.com/drive/folders/15xspHQQyw9lEcqJw3HLwf7-LvgNU7Ijc

2 Methodology

The audio mosaicing technique is comprised of two distinct components: a target audio and a set of source audio. To decide how the source audio should be split and combined to reconstruct the target, several key decisions must be made:

- how to slice the target audio
- how to slice the source audio
- which characteristics of the target to emulate
- how to select frames from the source audio
- how to recombine the selected source frames

In the following sections I go through these concerns individually and describe the strategies I used at each step.

2.1 Target sound selection

One of the first steps in audio mosaicing is selecting the target sound we want to emulate the characteristics of. I am attempting to recreate the melodies of popular songs, so I choose musical tracks with well-known melodies as targets. In particular, I chose two tracks with simple melodies that are easy to sing along to. The tracks are listed in Table 1 along with the timestamps used from each. Both clips have a vocal melody that I hope to capture.

[1] https://github.com/gulnazaki/meowify

[2] https://www.youtube.com/watch?v=8xiwuumLkOQ

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

Table 1: Target songs

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

2.2 Target characteristics

The next concern in audio mosaicing is to define the characteristics of the target sound we want to preserve. My 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, I used the the MELODIA algorithm introduced by Salamon and Gómez [2012] and implemented in the Essentia audio analysis toolkit [Bogdanov et al., 2013].

I subsequently used the pitch contour segmentation algorithm in Essentia introduced by McNab et al. [1995] to determine the pitch onset times. This segmentation algorithm is especially convenient for my needs as it takes as input an extracted pitch contour, 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 I chose relatively small values to ensure it captures the rhythmic nuance of the melodies. Specifically, I derived the minimum note duration using the tempo of the target tracks together with shortest note values used in their melodies. For example, 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 I set the "minDuration" parameter to just below this value to account for some variation in tempo. I set the "rmsThreshold" parameter to be low as well because I think over-detecting onsets is 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.

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. I created a set of audio from animals whose sounds tend to be short and clearly recognizable. I made sure to choose a set of animals with distinct timbres from one another in hopes of adding a bit of dynamism to the result. The parameters I used to query Freesound for the sound collection are shown in Table 2. In an effort to get improved quality sounds each query also included a sort condition by descending rating to get the highest-rated sounds first.

I chose a large selection of sounds to increase the chance of having high coverage of the melodic pitch range. For the same reason I also chose animals whose vocal cords produce sounds in different registers, focusing on those that overlap with the range of the human voice. With the exception of the moos and bleats, each collection was filtered to be less than a few seconds to increase the chances that the query would return single-shot sounds. In a manual exploration of Freesound query results, I noticed shorter recordings were typically of higher quality than longer field recordings which were often clouded with background noise and non-animal sounds. The moos and bleats were allowed to be slightly longer because I think cows and sheep take longer to vocalize their thoughts.

I also created a second source collection comprised of violin sounds with the hypothesis that a set of tuned, harmonic sounds would help make clear how well the technique was actually working. To create this collection, I filtered for single events using the "ac_single_event" query parameter in the Freesound API to specify that the audio file should contain one single audio event, i.e. a single note played on the violin. I queried for 100 such sounds to try and attain a wide coverage of pitch range.

Count	Keyword	Max duration (seconds)
5	horse whinny	5
5	horse neighing	5
10	cat meow	5
20	dog bark	1
20	bleat	10
20	cow moo	10
20	bird chirp	10

Table 2: Query parameters used in Freesound API requests

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. Among the top 10 highest scoring source frames, I choose one randomly so as to add a bit more variation in the result.

I experimented with several sets of features for computing this similarity:

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

As previously mentioned, the pitch values were estimated from the MELODIA algorithm in Essentia. 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. I tried computing similarity with and without MFCCs because I expected timbre to be less important than pitch in producing a convincing result. I 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

Finally, a strategy must be defined for how the selected source frames will be combined to reconstruct the target audio. I chose to recombine the selected source frames at the pitch onsets determined by the segmentation algorithm. I elected to not allow the overlapping of source frames during recombination.

3 Results

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

`https://drive.google.com/drive/folders/19LQ5WjfAgDFgQrcgZRImYsVU6Sk6d9TS.`

4 Discussion

Overall, the methodology described in this paper has a ways to go to produce the animal symphonies that I originally envisioned. I think the biggest improvement would come from assembling a well-curated sound collection. One glaring issue in my current animal sound collection is that many of the returned query results were actually sounds that were orthogonal to the query. For example, my query for "cat meow" returned 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. Additionally, the audio quality of the sounds is quite spurious. 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 to help filter out sounds with a lot of background noise like static or wind. A "harmonic" attribute would also be helpful in determining whether a sound is suitable to be used to match any pitch.

Another major improvement could come by using octave reduction. Some source frames match pitches in the target melodies but in a different octave. Labeling each source frame with its pitch class rather than its frequency would increase the coverage of pitch classes that can be matched exactly.

While the pitch detection algorithm I used worked well on the target sounds I chose, there are many complex mixtures on which it does not work well. For these cases, a source separation pre-processing step similar to the one used by Meowify can be used to isolate the instrument playing the predominant melody, so as to simplify the input for pitch detection. To keep the scope of the prototype small I did not implement such a source separation step.

5 Future Experiments

I am curious to try to perform audio mosaicing on some non-melodic target sounds. In particular I am keen to use the technique to emulate human speech, with a focus on phonetics and prosody characteristics. I am reminded of *Prisencolinensinainciusol*[3], a performance of a non-English language that sounds very much like English. It seems possible to use mosaicing with speech for both the target and source collection to create similarly disorientingly realistic non-speech speech. I wonder if the result would end up using phonemes and truncated syllables from the source sounds, or recreating the same sentence where each word comes from a different speaker, or stringing English words together that don't semantically make sense, e.g. "The running black hi ball joust your back home is back to the left left in left of your pocket bag and left..."

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.

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

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.

[3] `https://www.youtube.com/watch?v=-VsmF9m_Nt8`