Transposer

CS2108 Mini Project

A0140036X Joel Lim
A0155950B Jeremiah Ang

Introduction

Problem

At a karaoke session

- Users have different voice range
- Karaoke softwares offer transposition feature
- Users do not know how much to transpose

Solution

An application that estimates key of a user’s singing of a song and transposes the original soundtrack to match it.

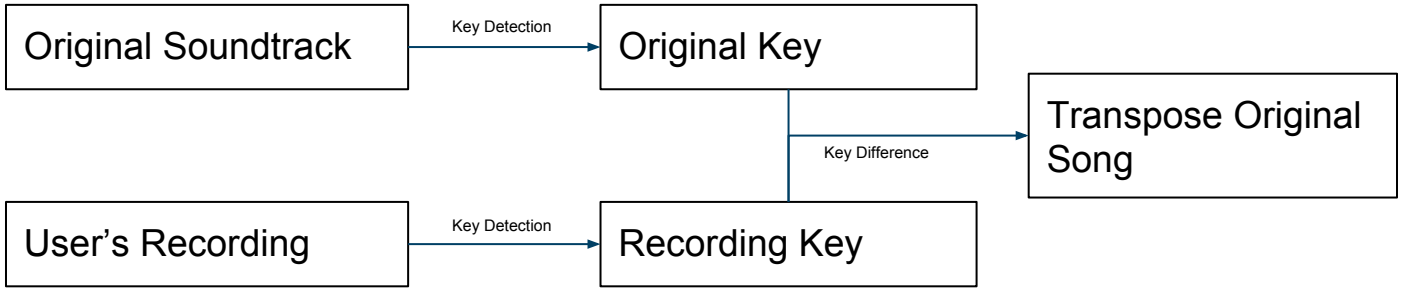
Background

Pitch, Keys and Transposition

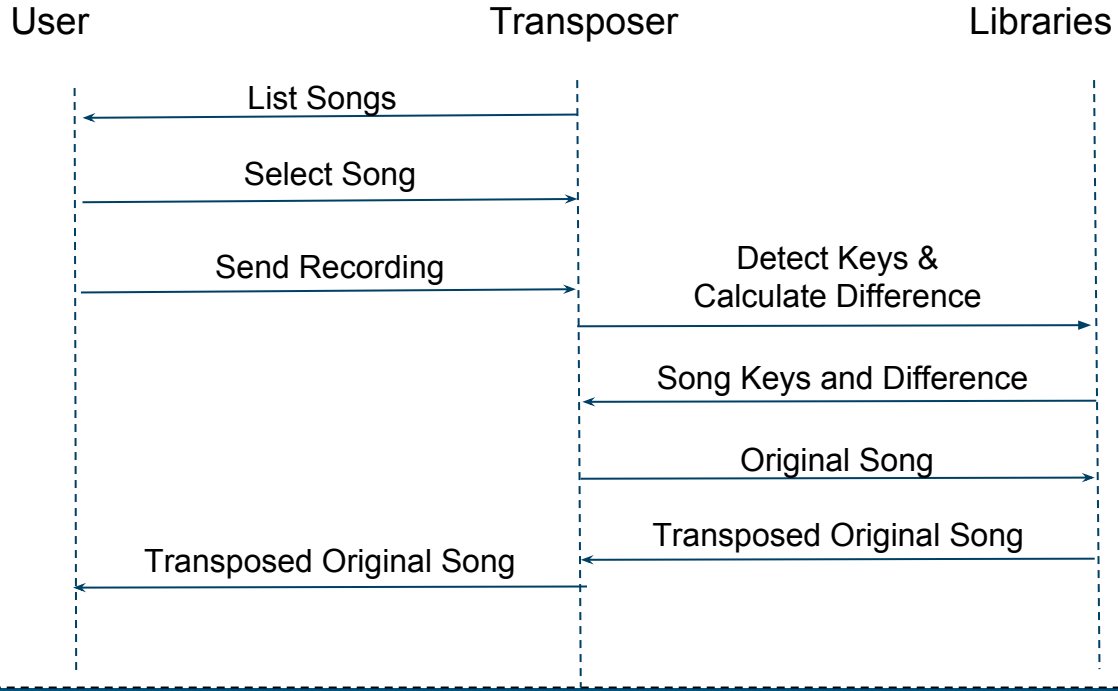
- *Pitch classes* are lettered (A, A#, B, C, ...), each a *step* from each other
- *Keys* are combinations of pitch classes
  - { C, D, E, F, G, A, B }, { C#, D#, F, F#, G#, A#, C }
  - *Root* note is the first note
  - *Scale degree* is index of pitch class in the scale
- Different keys played together sound unpleasant
- Transposition
  - Shifting a sequence of notes step-wise

Design

Flow


Original Soundtrack
Key Detection
Original Key
Transpose Original Song
Key Difference
User's Recording
Key Detection
Recording Key

User Interface



Implementation

Transposition

- librosa

Transposition

- Shift pitch of a song with librosa
- Apply *pitch-shift* to a waveform by *n* steps

```
def transpose (src, out, steps):
  y, sr = librosa.load(src)
  y_third = librosa.effects.pitch_shift(y, sr, steps)
  librosa.output.write_wav(out, y_third, sr)
```

File Conversion

- wav/mp3 to midi
  - audio_to_midi_melodia (melody)

audio_to_midi_melodia (melody)

- Converts wav/mp3 to midi with Melodia
- Extracts tempo with librosa

```
def audio_to_bpm(infile):
    y, sr = librosa.load(infile)
    tempo, beats = librosa.beat.beat_track(y=y, sr=sr)
    return int(tempo)
```

audio_to_midi_melodia (melody)

- Converts wav/mp3 to midi with Melodia
- Extracts tempo with librosa
- Extracts melody notes from an audio file based on given tempo and exported as MIDI

```
bpm = audio_to_bpm(filepath)
audio_to_midi_melodia(filepath, output_filepath, bpm)
```

Key Detection

- midi_properties

midi_properties

- Simple algorithm
- Analogous to text information retrieval
  - Keys vs Documents
    - Pitch Class vs Term
  - Note sequences vs Query
- Weights are given to scale degrees
- Count occurrences of pitch classes in note sequence
- Estimated key is key with highest similarity to note sequences

midi_properties - Keys

Keys

Scale Degree / Key	1 Notes	2	3	4	5	6	7
C	C	D	E	F	G	A	B
C#	C#	D#	F	F#	G#	A#	C
D	D	E	F#	G	A	B	C#
D#	D#	F	G	G#	A#	C	D
E	E	F#	G#	A	B	C#	D#
F	F	G	A	A#	C	D	E
F#	F#	G#	A#	B	C#	D#	F
G	G	A	B	C	D	E	F#
G#	G#	A#	C	C#	D#	F	G
A	A	B	C#	D	E	F#	G#
A#	A#	C	D	D#	F	G	A
B	B	C#	D#	E	F#	G#	A#

midi_properties - Vector Representation

Vector (weights)

Pitch Class Key	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	10	0	2	0	10	4	0	2	0	2	0	6
C#	6	10	0	2	0	10	4	0	2	0	2	0
D	0	6	10	0	2	0	10	4	0	2	0	2
D#	2	0	6	10	0	2	0	10	4	0	2	0
E	0	2	0	6	10	0	2	0	10	4	0	2
F	2	0	2	0	6	10	0	2	0	10	4	0
F#	0	2	0	2	0	6	10	0	2	0	10	4
G	4	0	2	0	2	0	6	10	0	2	0	10
G#	10	4	0	2	0	2	0	6	10	0	2	0
A	0	10	4	0	2	0	2	0	6	10	0	2
A#	2	0	10	4	0	2	0	2	0	6	10	0
B	0	2	0	10	4	0	2	0	2	0	6	10

midi_properties - Result

```
File data/F-Somebody_That_I_Used_To_Know_vocals_original_melodia.mid
*****************************************************************

Pitch Class             | C     C#      D       D#      E       F       F#      G       G#      A       A#      B
Pitch Counts            | 30.0  14.0    42.0    8.0     64.0    90.0    80.0    118.0   60.0    96.0    38.0    8.0
Sanitized Counts        | 0     0       42.0    0       64.0    90.0    80.0    118.0   60.0    96.0    38.0    0
Sanitized pitches       | -     -       D       -       E       F       F#      G       G#      A       A#      -

Results
F       | 0.765847685473
G       | 0.624596239161
D       | 0.582662216037
D#      | 0.57052341987
G#      | 0.534107031367
A       | 0.533003504443
F#      | 0.528589396746
A#      | 0.519761181351
E       | 0.517554127503
C       | 0.513140019805
C#      | 0.483344792849
B       | 0.262639407986
```

Demo

Demonstration

- Notes matching
  - E note.wav
- Key matching
  - C scale.wav
- Song matching
  - 小薇（黄品源)

Application Strategies

- Original Song Key Detection
- Cached Storage

Original Song Key Detection

- Key Detection known to be an estimate
- Inaccurate key guess = Inaccurate key difference
- Manual checking and input of original song key

Cached Storage

- Transposition of entire music file takes time
- Files are pre-transposed and stored

midi_properties

- Simple algorithm
- Analogous to text information retrieval
  - Keys vs Documents
    - Pitch Class vs Term
  - Note sequences vs Query
- Weights are given to scale degrees
- Estimated key is key with highest similarity to note sequences

Conclusion

Learnings

- Explored various concepts and technologies
  - Note Detection
  - Key Detection
  - File Format Conversion
  - Transposition
- Viabilities
- Current state of research and technologies

Challenges and Constraints

- Key Detection
  - Result is only an estimation
  - Based on many assumptions
    - Theoretical Model
    - Genre
    - Song structure
- Transposition
  - Getting transposed track to sound natural

Improvements/ Applications

- Use assumptions about domain to get better key estimates
  - Pitch range of melody in song or section
  - Patterns in melody
- Transpose in real time as user is singing
- Help musicians and singers determine keys of songs to be played in

QA