Automated Transcription of a Lyric's Melody

David Branner

Hacker School, New York

20141023

Automated* Transcription of a Lyric's Melody

David Branner

Hacker School, New York
20141023

Automated* Transcription of a Lyric's Melody

(*if you've done a lot of preliminary manual work)

David Branner

Hacker School, New York

20141023

In September I reported on an experiment to transcribe the solo singing voice with automated tools.

In September I reported on an experiment to transcribe the solo singing voice with automated tools.

I have recordings that I want to study — specifically, I want to study how the melody reflects the organization of the words that the music is set to.

In September I reported on an experiment to transcribe the solo singing voice with automated tools.

I have recordings that I want to study — specifically, I want to study how the melody reflects the organization of the words that the music is set to.

I want the text, annotated with its melody. A normal musical transcription supplies the opposite of that: a melody annotated with words.

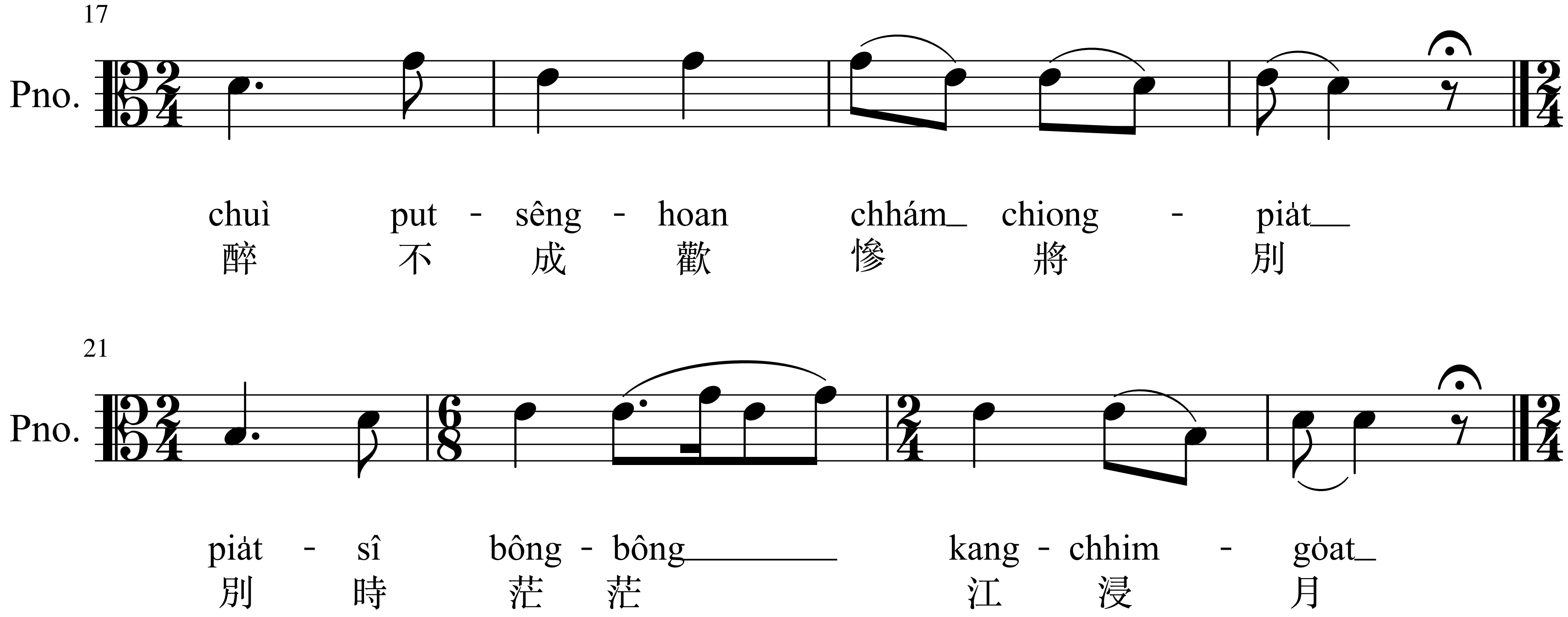
My earlier transcription (audio => MIDI => Python editing) went very badly because of tremolo in the singer’s voice and reverberation in the recording.

My earlier transcription (audio => MIDI => Python editing) went very badly because of tremolo in the singer’s voice and reverberation in the recording.

In the end I transcribed the piece by hand, using an open-source program called MuseScore.

My earlier transcription (audio => MIDI => Python editing) went very badly because of tremolo in the singer’s voice and reverberation in the recording.

In the end I transcribed the piece by hand, using an open-source program called MuseScore. Example:


17
Pno.
chuì put - sêng - hoan chhám_ chiong - piȧt__
醉 不 成 歡 慘 將 別
21
Pno.
piȧt - sî bông - bông________ kang - chhim - gȯat_
別 時 茫 茫 江 浸 月

Initially I was pleased because MuseScore can play back the melody I transcribe, speeding the work.

Initially I was pleased because MuseScore can play back the melody I transcribe, speeding the work.

But I was in luck because it can also export the score to an open-source format called MusicXML.

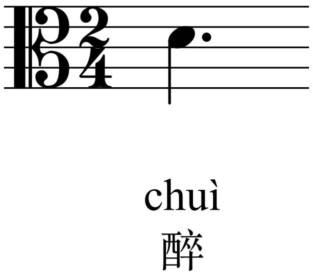
Initially I was pleased because MuseScore can play back the melody I transcribe, speeding the work.

But I was in luck because it can also export the score to an open-source format called MusicXML.

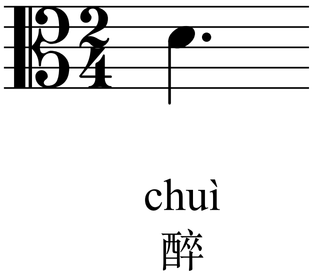
This turned out to allow me to do exactly what I had originally wanted — to study the melody of each syllable, quantitatively.

Here is a single note of the 1100-odd note performance I transcribed (the text is Taiwanese in Church *romaji*):

Here is a single note of the 1100-odd note performance I transcribed (the text is Taiwanese in Church *romaji*):

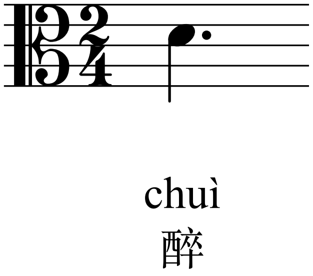


Here is a single note of the 1100-odd note performance I transcribed (the text is Taiwanese in Church *romaji*):



and here is the XML for that note:

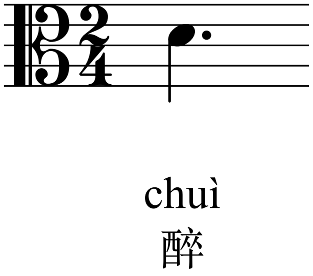
Here is a single note of the 1100-odd note performance I transcribed (the text is Taiwanese in Church *romaji*):



and here is the XML for that note:

```
<measure number="17">
  <attributes>
    <time>
      <beats>2</beats>
      <beat-type>4</beat-type>
      </time>
    </attributes>
  <note>
    <pitch>
      <step>D</step>
      <octave>4</octave>
      </pitch>
    <duration>12</duration>
    <voice>1</voice>
    <type>quarter</type>
    <dot/>
    <stem>down</stem>
    <lyric number="1">
      <syllabic>single</syllabic>
      <text font-family="Times New
Roman">chuì</text>
      </lyric>
    <lyric number="2">
      <syllabic>single</syllabic>
      <text font-family="Times New
Roman">醉</text>
      </lyric>
    </note>
```

Here is a single note of the 1100-odd note performance I transcribed (the text is Taiwanese in Church *romaji*):



and here is the XML for that note:

```
<measure number="17">
  <attributes>
    <time>
      <beats>2</beats>
      <beat-type>4</beat-type>
      </time>
    </attributes>
  <note>
    <pitch>
      <step>D</step>
      <octave>4</octave>
      </pitch>
    <duration>12</duration>
    <voice>1</voice>
    <type>quarter</type>
    <dot/>
    <stem>down</stem>
    <lyric number="1">
      <syllabic>single</syllabic>
      <text font-family="Times New
Roman">chuì</text>
      </lyric>
    <lyric number="2">
      <syllabic>single</syllabic>
      <text font-family="Times New
Roman">醉</text>
      </lyric>
    </note>
```

That’s exactly what I need to turn

That’s exactly what I need to turn words set to music (traditional score-notation)

That’s exactly what I need to turn
words set to music (traditional score-notation)
into

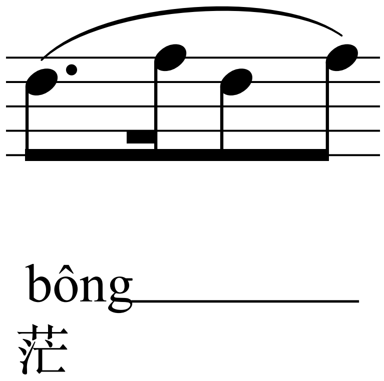
That’s exactly what I need to turn
    words set to music (traditional score-notation)
into
    melody on each syllable.

That’s exactly what I need to turn
    words set to music (traditional score-notation)
into
    melody on each syllable.

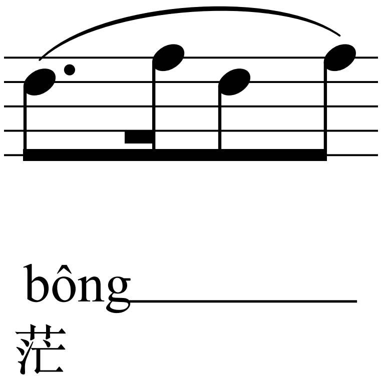
The point is to have a representation of the melody organized by words, rather than the other way around.

Another example:

Another example:

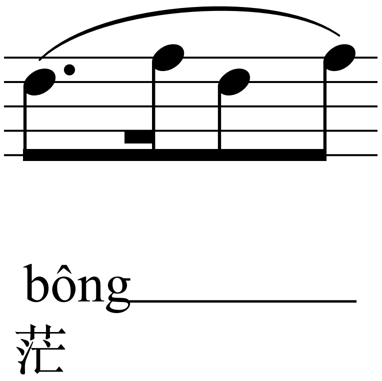


Another example:



```
<note>
  <pitch>
    <step>E</step>
    <octave>4</octave>
    </pitch>
  <duration>6</duration>
  <voice>1</voice>
  <type>eighth</type>
  <dot/>
  <stem>down</stem>
  <beam number="1">begin</beam>
  <notations>
    <slur type="start" number="1"/>
    </notations>
  <lyric number="1">
    <syllabic>end</syllabic>
    <text font-family="Times New Roman">bông</text>
    </lyric>
  <lyric number="2">
    <syllabic>single</syllabic>
    <text>茫</text>
    </lyric>
  </note>
<note>
  <pitch>
    <step>G</step>
    <octave>4</octave>
    </pitch>
  <duration>2</duration>
  <voice>1</voice>
  <type>16th</type>
  <stem>down</stem>
  <beam number="1">continue</beam>
  <beam number="2">forward hook</beam>
  </note>
<note>
  <pitch>
    <step>E</step>
    <octave>4</octave>
    </pitch>
  <duration>4</duration>
  <voice>1</voice>
  <type>eighth</type>
  <stem>down</stem>
  <beam number="1">continue</beam>
  </note>
<note>
  <pitch>
    <step>G</step>
    <octave>4</octave>
    </pitch>
  <duration>4</duration>
  <voice>1</voice>
  <type>eighth</type>
  <stem>down</stem>
  <beam number="1">end</beam>
  <notations>
    <slur type="stop" number="1"/>
    </notations>
  </note>
```

Another example:



```
<note>
  <pitch>
    <step>E</step>
    <octave>4</octave>
    </pitch>
  <duration>6</duration>
  <voice>1</voice>
  <type>eighth</type>
  <dot/>
  <stem>down</stem>
  <beam number="1">begin</beam>
  <notations>
    <slur type="start"
number="1"/>
    </notations>
  <lyric number="1">
    <syllabic>end</syllabic>
    <text font-family="Times
New Roman">bông</text>
    </lyric>
  <lyric number="2">
    <syllabic>single</syllabic>
    <text>茫</text>
    </lyric>
  </note>
<note>
  <pitch>
    <step>G</step>
    <octave>4</octave>
    </pitch>
  <duration>2</duration>
  <voice>1</voice>
  <type>16th</type>
  <stem>down</stem>
  <beam number="1">continue</
beam>
  <beam number="2">forward
hook</beam>
  </note>
<note>
  <pitch>
    <step>E</step>
    <octave>4</octave>
    </pitch>
  <duration>4</duration>
  <voice>1</voice>
  <type>eighth</type>
  <stem>down</stem>
  <beam number="1">continue</
beam>
  </note>
<note>
  <pitch>
    <step>G</step>
    <octave>4</octave>
    </pitch>
  <duration>4</duration>
  <voice>1</voice>
  <type>eighth</type>
  <stem>down</stem>
  <beam number="1">end</beam>
  <notations>
    <slur type="stop"
number="1"/>
    </notations>
  </note>
```

It is easy to parse the XML using the `lxml` library; `lxml`’s `root.xpath` method allows me request all the `note` elements as a generator, and I can step through the generator’s output, identifying syllables and then retrieving the melody to assigned to those syllables.

Important edge cases:

Important edge cases:

1. rests — they have no pitch content; also, since they don’t correspond to syllables, consecutive rests can be collapsed into one;

Important edge cases:

1. rests — they have no pitch content; also, since they don’t correspond to syllables, consecutive rests can be collapsed into one;

2. notes tied (but not slurred) together — since they correspond to one or part of one syllable and are all the same pitch, they can be collapsed into one;

Important edge cases:

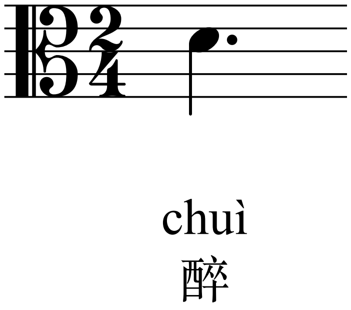
1. rests — they have no pitch content; also, since they don't correspond to syllables, consecutive rests can be collapsed into one;

2. notes tied (but not slurred) together — since they correspond to one or part of one syllable and are all the same pitch, they can be collapsed into one;

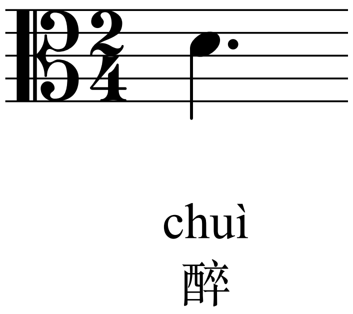
3. melisma — more than one note per vowel or syllable — represented as a series of notes on one syllable.

Output of the first note above:

Output of the first note above:



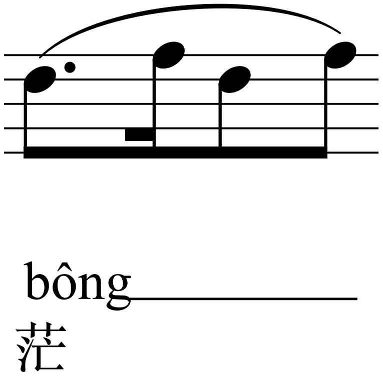
Output of the first note above:



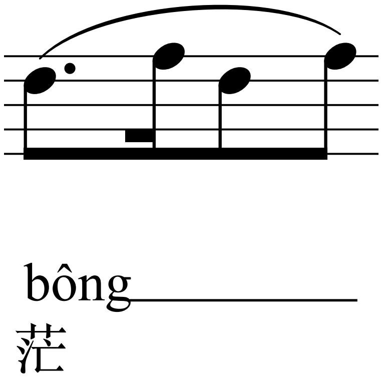
```
...
('chuì',
  [{'pitch_data': {'octave': '4', 'step': 'D'},
    'lyric_2': {'syllabic': 'single', 'text': '醉'},
    'duration': 12}]),
...
```

Output of the more melismatic second example above:

Output of the more melismatic second example above:

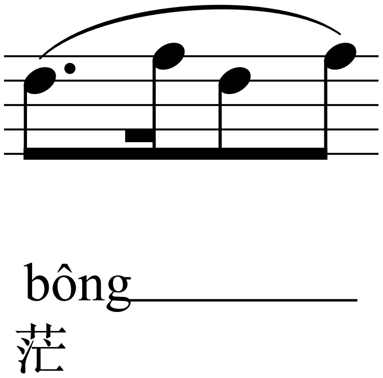


Output of the more melismatic second example above:



```
...
('bông',
  [{'pitch_data': {'octave': '4', 'step': 'E'},
    'lyric_2': {'syllabic': 'single', 'text': '茫'},
    'duration': 6},
   {'pitch_data': {'octave': '4', 'step': 'G'}, 'duration': 2},
   {'pitch_data': {'octave': '4', 'step': 'E'}, 'duration': 4},
   {'pitch_data': {'octave': '4', 'step': 'G'}, 'duration': 4}]),
...
```

Output of the more melismatic second example above:



```
...
('bông',
  [{'pitch_data': {'octave': '4', 'step': 'E'},
    'lyric_2': {'syllabic': 'single', 'text': '茫'},
    'duration': 6},
   {'pitch_data': {'octave': '4', 'step': 'G'}, 'duration': 2},
   {'pitch_data': {'octave': '4', 'step': 'E'}, 'duration': 4},
   {'pitch_data': {'octave': '4', 'step': 'G'}, 'duration': 4}]),
...
```

Final output is list of tuples: [(syllable, [note-dicts]), ...].

Final output is list of tuples: [(syllable, [note-dicts]), ...].

Next steps:

Final output is list of tuples: [(syllable, [note-dicts]), ...].

Next steps:

1. Conversion of `'octave'` and `'step'` attributes of `'pitch_data'` to MIDI pitch numbers.

Final output is list of tuples: [(syllable, [note-dicts]), ...].

Next steps:

1. Conversion of `'octave'` and `'step'` attributes of `'pitch_data'` to MIDI pitch numbers.

2. Analysis of the melodic values of syllables and language phrases.

End