

Chapter 5

Results and Testing

This chapter will describe the results of my work with of W. A. Mozart's *Eine kleine Nachtmusik*, K.525 as an example. This chapter will also look at timing metrics of the system.

5.1 An Example: Eine kleine Nachtmusik, K.525, W. A. Mozart)

This section will go through an example of how the entire `OMRMIDICorrector` system works on W. A. Mozart's *Eine kleine Nachtmusik*, K.525.

Mozart
Eine Kleine Nachtmusik
K. 525

Allegro

Violine I

Violine II

Viola

Violoncello und Kontrabaß

Figure 5-1: The first page of the scanned copy of the score found on IMSLP.

5.1.1 Musical Properties of K.525

I chose K.525 as the example piece for many of its music properties that make it a difficult but interesting piece to align and correct. In the hashing, align

5.1.1.1 Bass and Cello Doubling

In the original scanned score, there are only four parts because the bass part doubles the cello part one octave lower. However, in the MIDI encoding, there must be five separate voices. Thus, in the preprocessing step of the `OMRMIDICorrector`, it would be ideal for the system to recognize this. Otherwise, in instances where the `OMRMIDICorrector` could not pair up the parts between the input OMR stream and the input MIDI stream, there would be an error thrown, and alignment and correction would not happen.

5.1.1.2 Tremolos

Starting in measure 5 in the original scanned score, the second violin has tremolo notes, instead of repeated 16th notes written out. The OMR parser in SmartScore is not robust enough to recognize the slashes across the stem of the note as tremolo markings and instead interprets it as a single note. The MIDI protocol has no way of indicating tremolos, so the second violin's notes in measure 5 are encoded as regular 16th notes. Ideally, the aligner would be robust enough to align tremolo measures with non-tremolo measures even though there will be a huge discrepancy in the number of notes present in the measure.

5.1.2 Preparing the Raw Input

The basic raw input of the system is an OMR score and a MIDI file. A scanned copy of the score was found in IMSLP [5], the source for much free public domain sheet music. The MIDI file was sourced from the Yale MIDI Archive.

Mozart
Eine Kleine Nachtmusik
K. 525

Allegro

Violine I
Violine II
Viola
Violoncello und Kontrabaß

Figure 5-2: The first page of the scanned copy of the score found on IMSLP.

I used SmartScore's built-in OMR tool to convert the scanned score of K.525 into an ENF file. Then I exported the post-OMR file as a musicXML file.

Similarly for the K.525 MIDI file, I used SmartScore to parse the original file and exported it as a musicXML file.

The last step of preparing the raw input is to parsing the musicXML files with the

music21 library so that they are `Stream` objects. This can be done by passing in the musicXML filepaths to the `parse` method in the `converter` class. We will call these two parsed streams `k525midiStream` and `k525omrStream`.

```
k+knfrom n+nnmusic21 k+knimport o*
nk525MidiFilepath o= l+s+s1'path_to_k525_mid_to_xml.xml'
nk525OmrFilepatho= l+s+s1'path_to_k525_omr_to_xml.xml'
nk525midistream o= nconvertero.nparsep(nk525MidiFilepathp)
nk525omrstream o= nconvertero.nparsep(nk525OmrFilepathp)
```

5.1.3 Running OMRMIDICorrector on K.525

5.2 Timing