Pattern recognition system
Exercise Project

Report Assignment n°2

Student
Antoine Honoré
honore@kth.se

Student
Audrey Brouard
brouard@kth.se

2015 Period 1

I

Below are represented the pitch and intensity profiles of the 3 songs provided. The window lentgh used to cut eh original signal into frames is 30 ms.

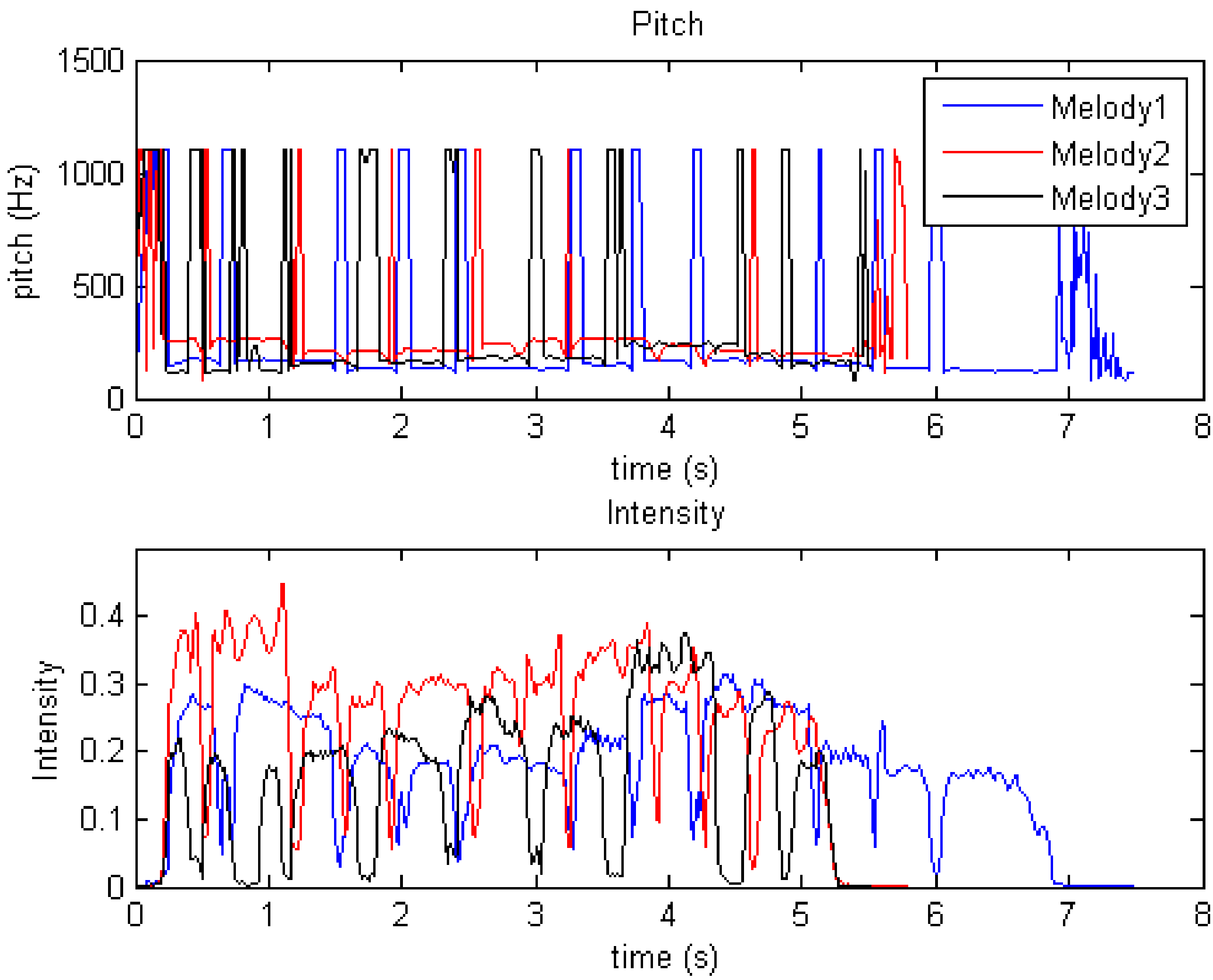

FIGURE 1 – Pitches and Intensity for the 3 melodies

On the pitch profile we can notice that we hardly see the different "useful" pitches as the noise has a pitch of around 1000Hz. The important thing here is that the pitch reaches 1000Hz with a very low intensity. Which means that a threshold on intensity should be enough to get rid of the noise. If we zoom in we can clearly read the pitch values corresponding to different notes :

Here on the pitch profile we can distinguish different values of pitch between 100Hz and 300Hz, which corresponds to different notes, globally spread over 1.5 octave.

II Design of the feature extractor

II.1 Our decisions

In order to fulfil all the conditions listed in the subject, we designed our feature extractor as follows :

- First we extract the actual pitches by removing the noise around 1000Hz so that we only have the pitches containing a lot of information.

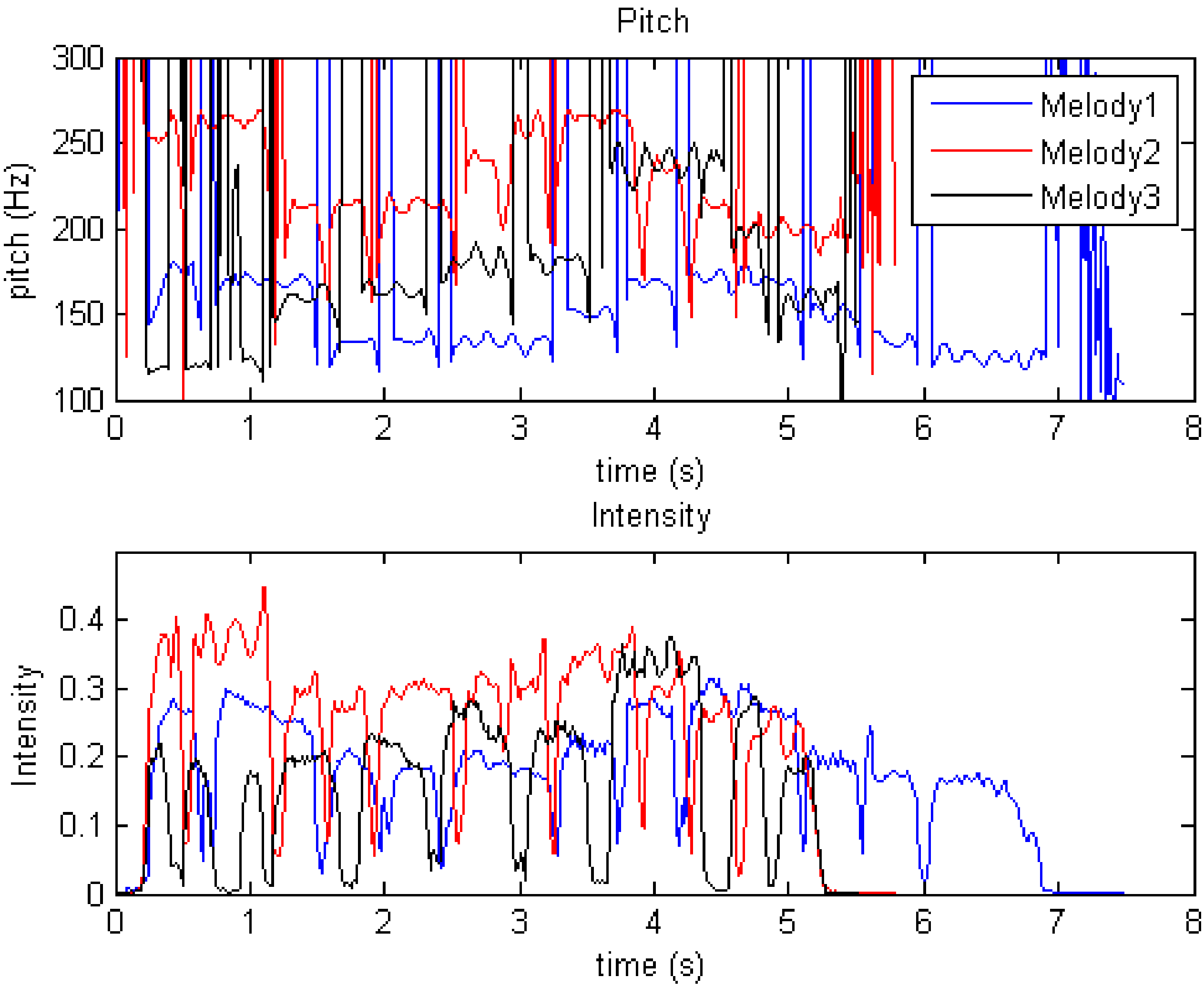

Figure 2 – Zoom

- Then we interpolate the vector thus obtained with a vector containing all the notes in the range 20Hz-20kHz. This last vector was built thanks to the constant ratio between 2 consecutive semitons. This way we obtain a vector containing the different notes played in the song, but so far we are dependant of the octave.
- To that purpose, we set a reference to remove the offset of the melody. The reference is defined as the first pitch which is not computed during a silence. After that, we are able to determine how many semitons above or below the reference the next pitches are located. We thus obtain a vector containing the variation of semiton from the reference for each pitch, and infinity (or NaN) when it's a silence.
- Finally, the duration of each note/silence will be determined by the number of time the source remains in the same state. To avoid temporal distorsion issues (a singer might sing faster or slower than another one), we will define a vector containing the ratio between two notes durations. (Actually we made some tests about that but had to drop them for the moment as it didn't work)

II.2 Integration to PattRecClasses

The states correspond to the semitons offset. Let us consider an example : we extract such features $[044440 - 1 - 1 \ldots]$. This vector is a vector of frames and here the first one is the reference. The five following fours mean that this part of the signal has a pitch 4 semitons above the reference. The states come naturally to be integers representing an offset of semitons. The actual value of pitches in frame 2 to 5 are not the same, their distribution defines the distribution of state 4. The extraction of the offset is made in the function find_offset.

II.3 Example

Let us now consider a real example. Melody 1 and melody 2 are the same melody sung at a different level. Besides, the two songs do not have the same duration. To perform the test we followed these steps :

- Load the two songs ;
- Run GetMusicFeatures with the default parameters on the two signals ;
- Run find_offset on the two signals and get v_1 & v_2, which represent the vectors of offset that we discussed about in the previous section.
  the plot of v_1 & v_2 is shown on figure 3.

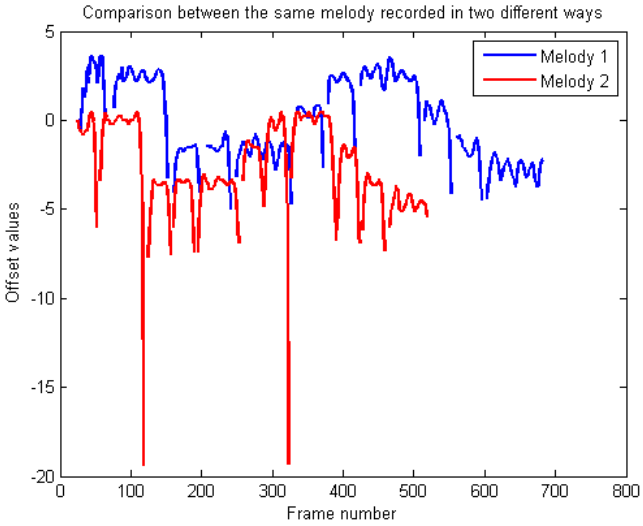

FIGURE 3 – Blue : melody 1 Red : melody 2

On the figure 3, we clearly see that the serie of semiton is the same. In the first melody, the singer hold the notes longer than in the second and our track of semitons keeps that information.

Now we try with a complete different song and we add to the graphics the third melody (figure 4), we can clearly see that the behaviour is completely different.

As the song sung is different, the notes are different and that is why the offset obtained changes. This is the behaviour we expected from our feature extractor.

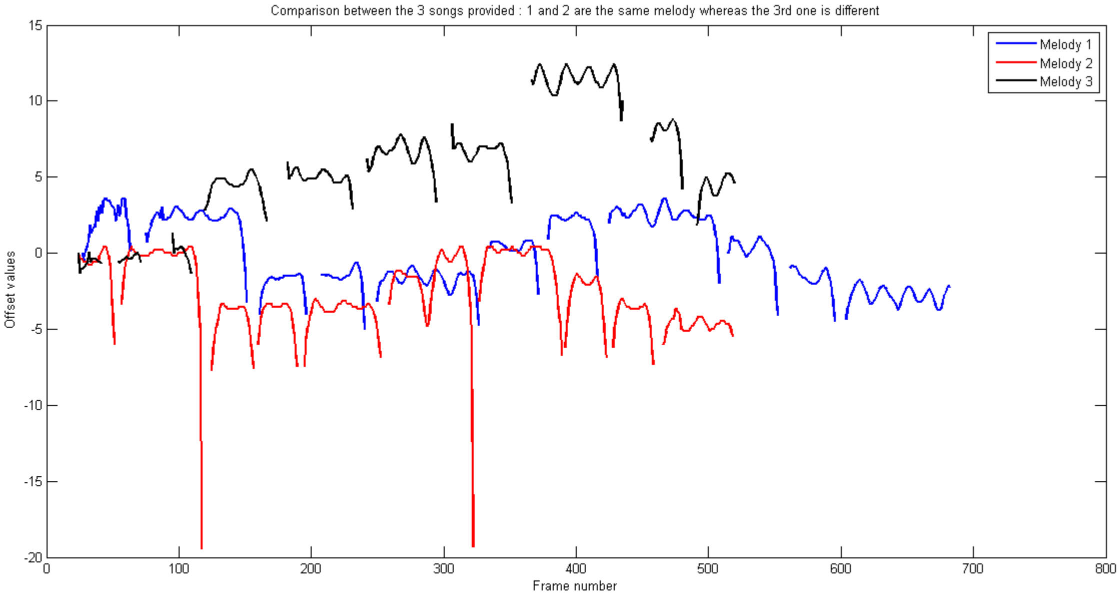

Figure 4 – Black : third melody, different from the 1 others
The black track decreases only at the end of the song

II.4 Weaknesses of our feature extractor

So far, our feature extractor can recognize a song sung even if it's transposed, thanks to the offset. To recognize two songs sang at different speed, we were thinking to rely on the state duration for the moment as our other solutions didn't come ton an end. However, for the moment our feature extractor doesn't take into account the ambient noise which can influence the pitch of the notes played. We tried to reproduce the melody 3 with our own voice. The result is shown on figure 5.

The feature extractor suffers from a lack of precision. It is very difficult to reproduce two same tracks of offset in normal[1] recording conditions. On figure 5 & 3, there is still a constant difference between the two tracks of semitons. Our reference was supposed to tackle this issue. The choice of the reference is to be reconsidered but so far we didn't come up with a better solution.

[1] Means with the microphone of a laptop and without being an excellent singer

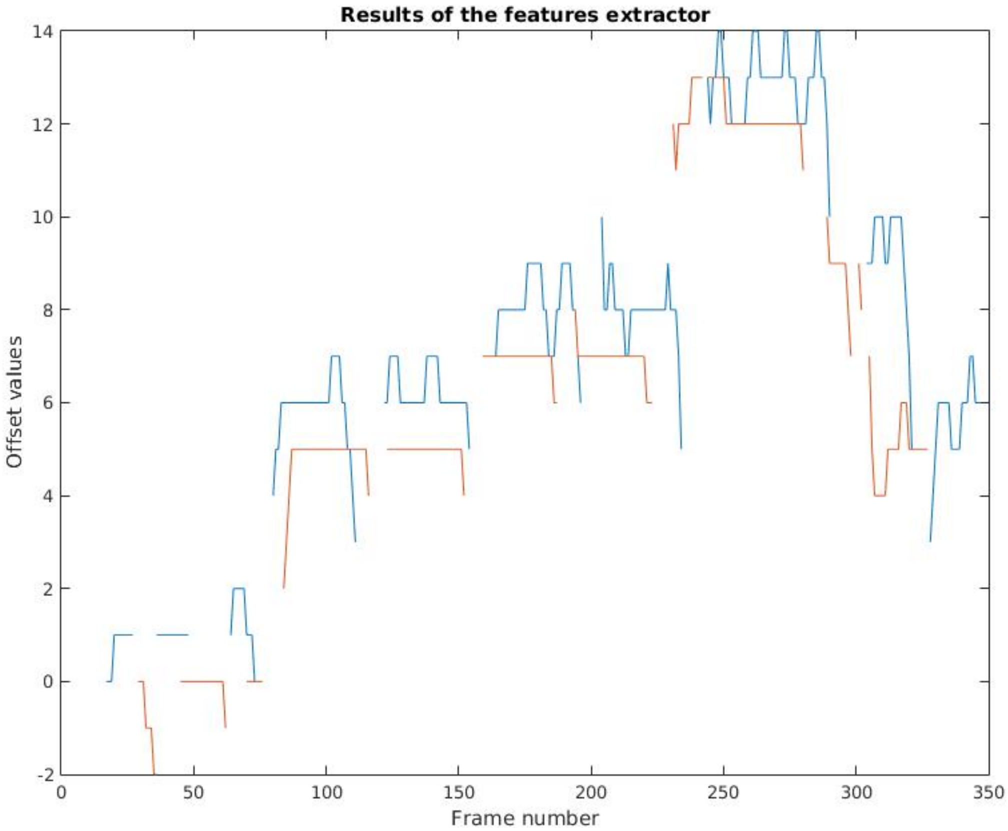

Figure 5 – Blue : Melody 3, Red : our own voice