FFT Workshop 2025

Yutao Liu s4213211
MSc Computer Science: Artificial Intelligence

October 2025

Assignment 1: Spectrogram

We computed the Short-Time Fourier Transform (STFT) of `piano.wav` using `librosa` and displayed the spectrogram. Figure 1 shows the result. The x-axis represents time (in seconds), the y-axis frequency (in Hz), and the color indicates the magnitude (yellow/red = higher energy, blue = lower energy). The horizontal harmonic lines visible in the figure correspond to overtones of individual piano notes.

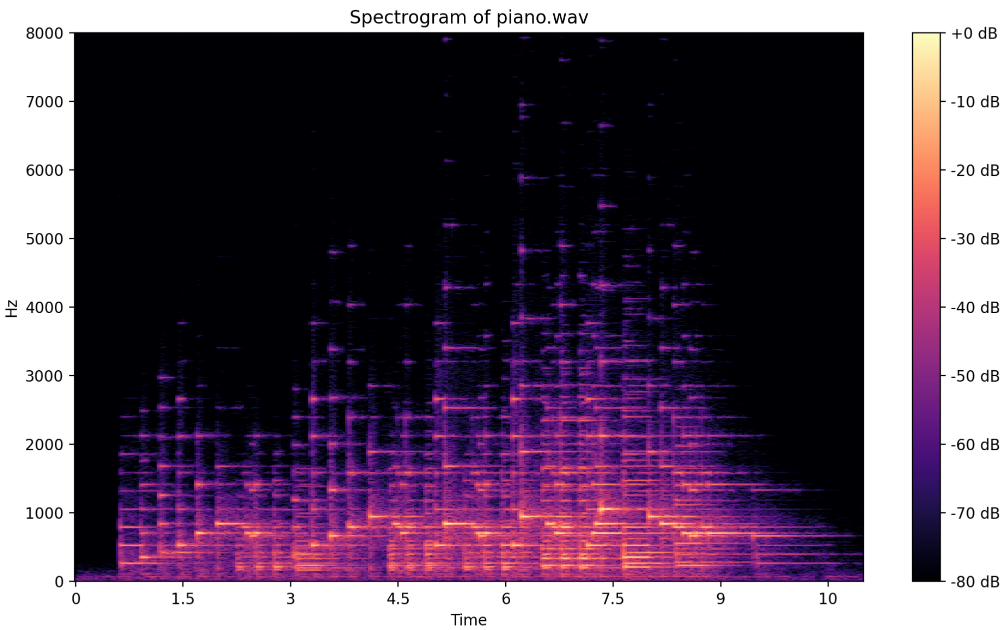

Figure 1: Spectrogram of `piano.wav`.

Assignment 2a: Feature Vectors

The signal was divided into windows of 512 samples (about 32 ms at 16 kHz), and the energy in eight frequency bands was computed:

$$[0, 1 \text{ kHz}), [1, 2 \text{ kHz}), \ldots, [7, 8 \text{ kHz})$$

The resulting vectors were stored in the file `piano_energies.txt`. Each row corresponds to one frame, with eight values representing the band energies. The energy per band was calculated as the sum of squared magnitudes within each frequency range of the FFT.

Assignment 2b: UDR Encoding

To improve melody recognition, we optimized the parameters as follows:

- **Window size:** 2048 samples (≈128 ms), hop size 512 samples (32 ms).
- **Number of bands:** 24, logarithmically spaced.
- **Frequency range:** 55 Hz–4186 Hz (A1–C8, covering the main piano range).

These choices increase frequency resolution, ensure bands align more naturally with musical notes, and remove unnecessary low/high frequencies. Using these parameters, the UDR encoding of `piano.wav` is shown below (compressed form):

R5 D1 U1 R9 U1 R9 D1 R3 D1 R2 U2 R2 D1 R3 U1 R4 D1 U1 R2 U1 R7 U1 R10 D1 U1 D1 R2 D1 U1 R2 D1 U1 R3 D1 R6 U1 R1 U1 R2 D1 R4 U1 R3 D1 U1 R2 U1 R7 U1 R8 U1 R8 D1 U2 R2 D1 R10 D2 R2 U1 R3 U1 R12 D1 R11 D1 R8 U1 R8 D1 U2 R2 D1 R9 D1 R3 U1 R2 U1 R1 U2 R4 D1 R4 D1 R10 U1 R11 D1 R6 D1 R1 U1 R24 D1 R19 D1 U2 R7 D1 R6 D1 R1 U1 R1 D1 R4 U1

The optimized encoding provides smoother U/D transitions and reduces spurious fluctuations compared with the default parameters.

Assignment 2c: Vader Jacob

We recorded the melody Vader Jacob and converted it to a mono 8-bit 16 kHz WAV file named mysong.wav. The same optimized parameters from Assignment 2b were used to process the recording.

The resulting UDR encoding (compressed form) is:

R2 U1 R4 D1 U1 R5 D1 U1 R5 D1 R3 U1 R6 D1 U1 R4 D1 R1 U1 R7 D1 R3 U1 R22 U1 R4 D1 U1 R6 D1 R2 D1 U1 R2 D1 U1 D1 R1 U1 D1 R5 U1 D1 R1 U1 R1 D1 R1 U1 R7 D1 R6 U1 R8 D1 U1 R8 U2 D1 R1 U1 R3 U1 R1 D1 R2 U1 D1 R1 U1 R1 U1 R3 U1 R1 D1 R4 D1 U1 R2 D1 R3 U1 D1 R1 U1 R1 D1 R2 U1 R2 D1 R2 U1 D1 R4 U1 R6 D1 U1 R8 U1 R1 D1 R4 U1 R2 D2 R7 U1 R2 D1 R3 U1 R2 D1 R8 U1 R2 U1 R11 D1 U1 D1 R1 U1 R8 U1 R9 U1 R6 U1 D1 R1 D1 U1 D1 R6 D2 R7 U1 R1 D1 R1 D1 R1 U2 R1 D2 U1 D1 R3 U1 R3 D1 R2 U2 D2 U1 R10 D1 R7 U1 R5 D1 R1 D1 U1 D1 R1 D1 R7 D1 R8 U1 R1 D1 R6 U1 D1 U1 R1 U1 R1 D1 R3 U1 D1 U1 R1 D1 U1 R2 D1 U1 R2 D1 U1 R1 D1 R1 U1 D1 R7 U2 D1 R2 U1 D1 U1 D1 R13 D1 R4 D1 R11 U1 R2 U1 R1 D1 R4 D1 R7 U1 R1 D1 R16 U1 D1 R18 U1 R18 U1 R5 U1 D1 R1 D1 R1 U1 R18

This result demonstrates that the UDR representation can also describe simple melodies such as "Vader Jacob", producing clear up–down–repeat patterns that follow the melodic contour.

Conclusion

The baseline configuration (512-sample windows, eight linear bands, 0–8 kHz) captured the overall spectral evolution but exhibited unstable UDR sequences with frequent jumps due to limited frequency resolution. By increasing the window size, applying logarithmic band spacing, and restricting the frequency range to the piano domain, the resulting UDR sequences became more stable and musically meaningful. Overall, this workshop demonstrates how careful parameter tuning in Fourier-based audio feature extraction improves the robustness and interpretability of symbolic melody encoding.