

# 計算機科学実験 4 音声 レポート 3

京都大学工学部情報学科

計算機科学コース 3 回生

学生番号: 1029-33-1415

氏名: 安済翔真

2024 年 1 月 11 日

## 目次

1	機能・プログラム説明	2
1.1	音量の表示 . . . . .	2
1.2	スペクトログラムの表示 . . . . .	2
1.3	音程の表示 . . . . .	2
1.4	音楽の再生・再生時間の表示 . . . . .	3
1.5	終了ボタン . . . . .	3
2	実行例とテスト	3
2.1	音量の表示 . . . . .	3
2.2	スペクトログラムの表示 . . . . .	4
2.3	音程の表示 . . . . .	5
2.4	音楽の再生・再生時間の表示 . . . . .	6
2.5	終了ボタン . . . . .	7
3	工夫点	7
3.1	音量の表示 . . . . .	7
3.2	音程の軸の表示 . . . . .	8
3.3	再生時間の表示 . . . . .	8
4	考察	8

# 1 機能・プログラム説明

## 1.1 音量の表示

マイクからの入力 of 音量を表示できるようにした。コード 1 は音量を表示するためのプログラムである。本課題ではストリーム処理を行うため、データ列の最後に現在のフレームの音量を追加している。

### 1 音量の表示

```
1 vol = 20 * np.log10(np.mean(x_current_frame ** 2) + EPSILON)
2 volume_data = np.roll(volume_data, -1)
3 volume_data[-1] = vol
```

## 1.2 スペクトログラムの表示

マイクからの音声のスペクトログラムを表示できるようにした。コード 2 はスペクトログラムを表示するためのプログラムである。音量が一定以下のときはスペクトログラムを表示しないようにしている。

### 2 スペクトログラムの表示

```
1 fft_spec = np.fft.rfft(x_stacked_data * hamming_window)
2 fft_log_abs_spec = np.log10(np.abs(fft_spec) + EPSILON)[-1]
3
4 spectrogram_data = np.roll(spectrogram_data, -1, axis=1)
5 spectrogram_data[:, -1] = fft_log_abs_spec
6 if vol < VOLUME_THRESHOLD:
7     spectrogram_data[:, -1] = np.array(fft_log_abs_spec) * np.nan
```

## 1.3 音程の表示

マイクからの音声の音程を表示できるようにした。コード 3 は音程を表示するためのプログラムである。音量が一定以下のときは音程を表示しないようにしている。

### 3 音程の表示

```
1 melody = shs(fft_log_abs_spec, SAMPLING_RATE, FRAME_SIZE)
2 melody_data = np.roll(melody_data, -1)
3 if vol < VOLUME_THRESHOLD:
4     melody = np.nan # 話していないときは音程をにする 0
5 melody_data[-1] = melody - NOTES[0]
```

## 1.4 音楽の再生・再生時間の表示

予め指定した音楽を再生できるようにした。コード 4 は音楽を再生するためのプログラムである。音楽の再生には pyaudio を用いている。

4 音楽の再生

```
1 p_play = pyaudio.PyAudio()
2 stream_play = p_play.open(
3     format = p.get_format_from_width(audio_data.sample_width),
4     channels = audio_data.channels,
5     rate = audio_data.frame_rate,
6     output = True
7 )
8 for chunk in make_chunks(audio_data, size_frame_music):
9     stream_play.write(chunk._data)
```

## 1.5 終了ボタン

アプリを終了できるボタンを配置した。コード 5 は終了ボタンを配置するためのプログラムである。終了ボタンを押すと画面が閉じるようになっている。

5 終了ボタン

```
1 button = tkinter.Button(master=root, text="終了", command=_quit, font=("", 30))
2 button.pack()
```

# 2 実行例とテスト

## 2.1 音量の表示

アプリを開くと図 1 のように音量が表示される。音量は右側の軸で表示されている。

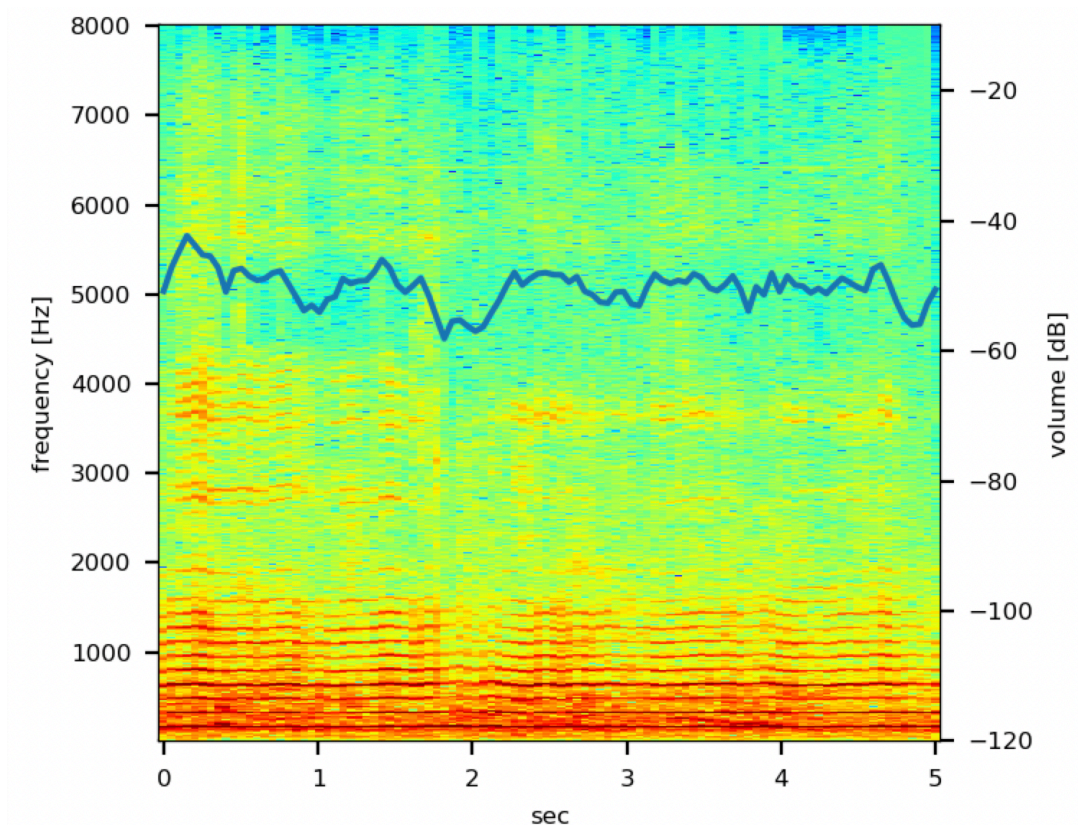


図1 音量の表示

## 2.2 スペクトログラムの表示

アプリと開くと図2のようにスペクトログラムが表示される。声を出していない区間ではスペクトログラムが表示されないことを確認した。

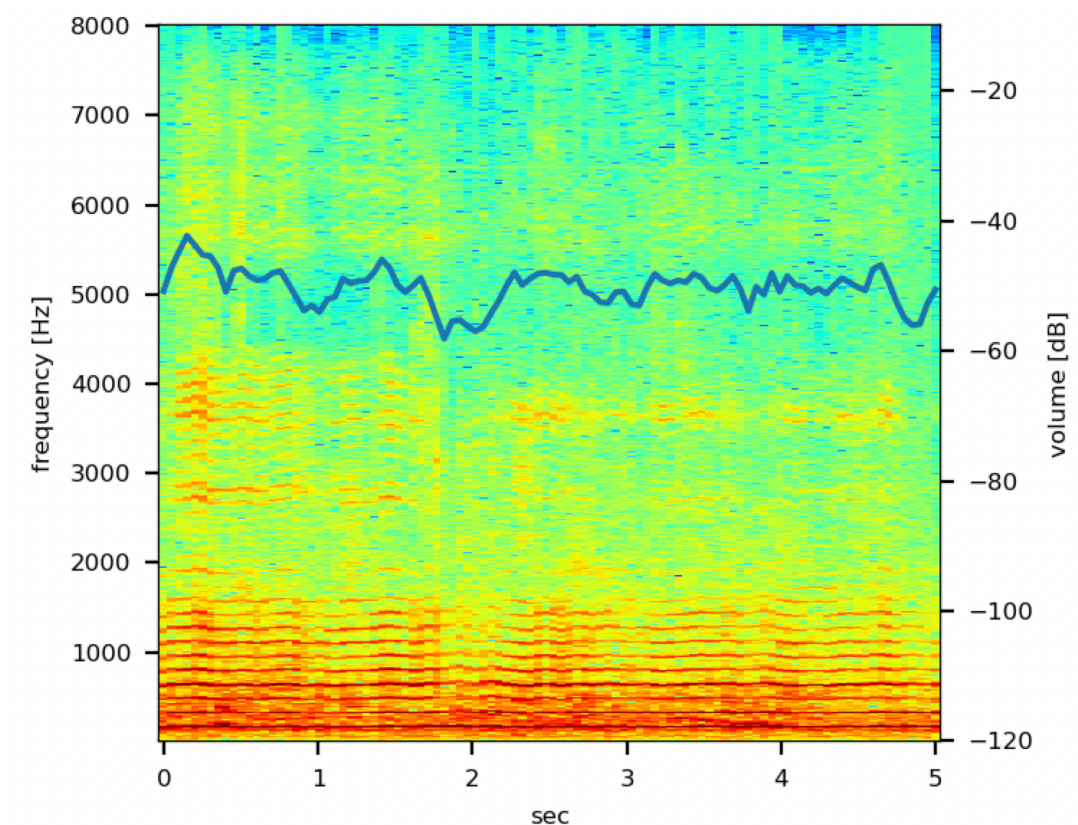


図2 スペクトログラムの表示

### 2.3 音程の表示

アプリを開くと図3のように音程が表示される。声を出していない区間では音程が表示されないことを確認した。

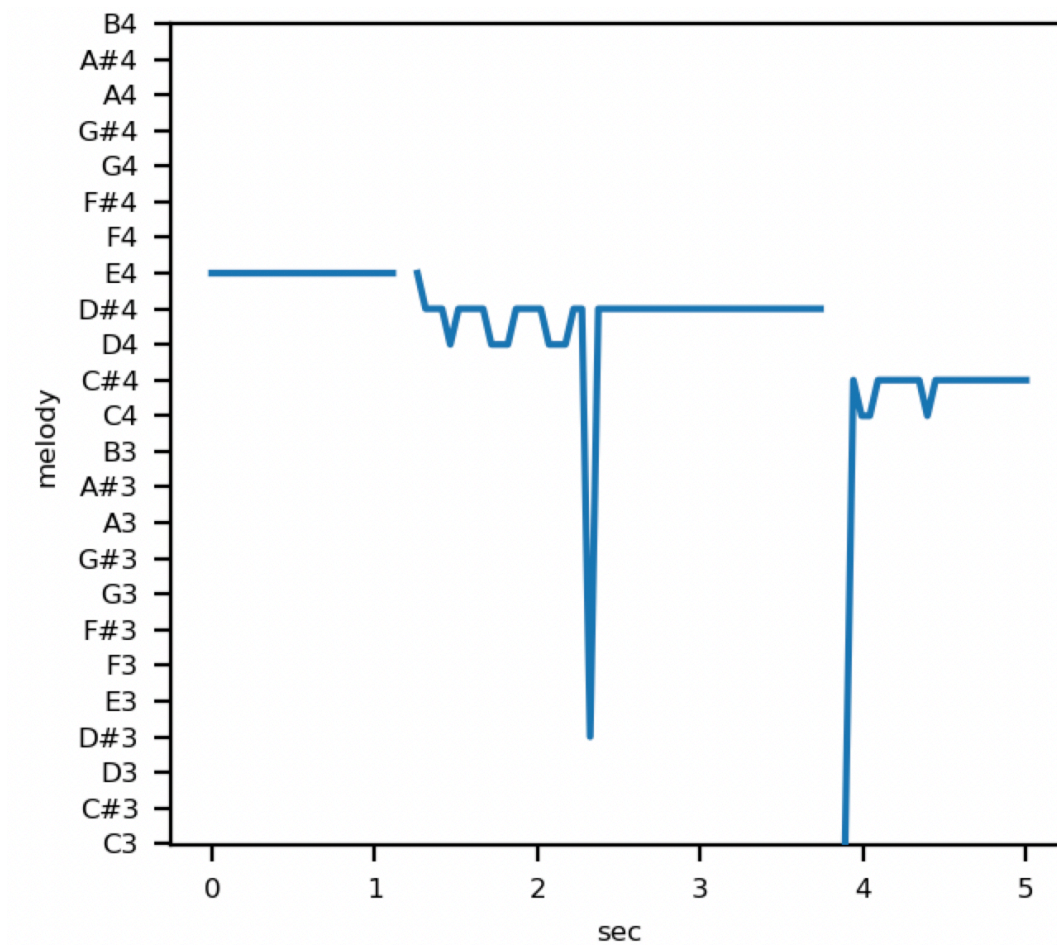


図3 音程の表示

## 2.4 音楽の再生・再生時間の表示

アプリを開くと音楽が再生される。また、図4のように再生時間が表示されることを確認した。



図4 音楽の再生・再生時間の表示

## 2.5 終了ボタン

アプリを開くと終了ボタンが表示される。終了ボタンを押すとアプリが終了することを確認した。



図5 終了ボタン

## 3 工夫点

### 3.1 音量の表示

マイクからの入力音声の音量を表示できるようにした。これにより、音の大きさを可視化し、音声の強弱を把握できるようになった。

### 3.2 音程の軸の表示

音程の軸を、ノート番号ではなく、音程の名前を表示できるようにした。(図 3) これにより、音程の把握が容易になった。

### 3.3 再生時間の表示

音楽の再生時間を表示できるようにした。これにより、歌い始めてからの経過時間を把握できるようになった。

## 4 考察

今回の課題では、入力音声のストリーム処理を行った。入ってくる入力を逐次処理するのは、状態の管理などが難しかったが良い経験になった。