



Convertisseur MIDI-WAV

Projet de Programmation 2022-2023



Objectifs

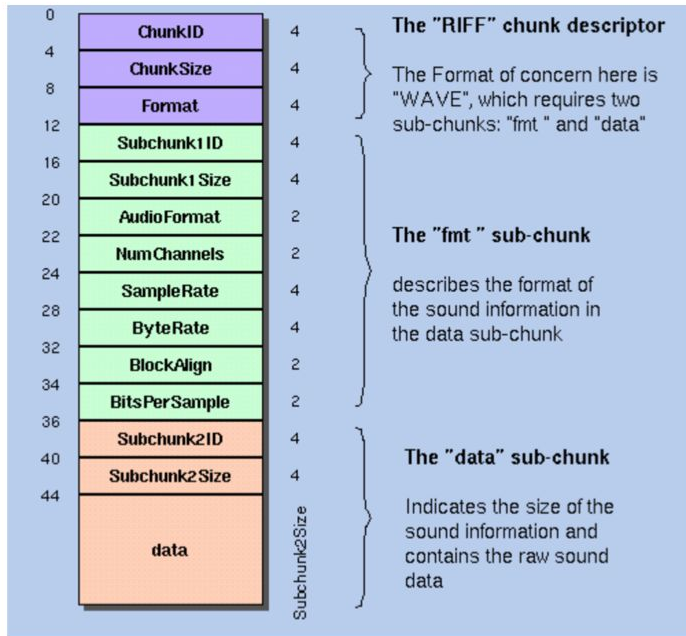
- Son monophonique

Fichier WAV



Fichier MIDI

WAV



MIDI

```

MThd
  chunk_size: 6
  format_type: 1
  nb_tracks: 2
  time-division: 480

MTrk
  length: 496
  meta-event
    v_time: 0
    type: 3 Sequence/Track Name
    v_length: 6
  meta-event
    v_time: 0
    type: 88 Time Signature
    v_length: 4
  meta-event
    v_time: 0
    type: 89 Key Signature
    v_length: 2
  meta-event
    v_time: 0
    type: 81 Tempo Setting
    v_length: 3
  midi-event
    v_time: 0
    status: 11 Controller
    channel: 0
    param1: 121
    param2: 0
  midi-event
    v_time: 0
    status: 12 Program Change
    channel: 0
    param1: 0
    param2: 0
  midi-event
    v_time: 0
    status: 11 Controller
    channel: 0
    param1: 7
    param2: 99
  midi-event
    v_time: 0
    status: 11 Controller
    channel: 0
    param1: 10
    param2: 63

midi-event
  v_time: 25
  status: 9 Note On
  channel: 0
  param1: 67
  param2: 90
midi-event
  v_time: 455
  status: 9 Note On
  channel: 0
  param1: 67
  param2: 0
midi-event
  v_time: 25
  status: 9 Note On
  channel: 0
  param1: 67
  param2: 90
...

MTrk
  length: 25
  meta-event
    v_time: 0
    type: 3 Sequence/Track Name
    v_length: 6
  meta-event
    v_time: 0
    type: 89 Key Signature
    v_length: 2
  meta-event
    v_time: 0
    type: 33
    v_length: 1
  meta-event
    v_time: 1
    type: 47 End Of Track
    v_length: 0
  
```

...

Difficultés de la conversion

Son monophonique : une voix

- Extraction des données .mid (parser)
- Détection des notes
- Détection de début et fin d'une note
- Gestion des silences

Résultats .mid -> .wav

Gamme do



Gamme avec silence au temps pair



Ode à la joie



Résultats .wav -> .mid (-> .wav)

Gamme de do



Gamme de do avec silence au temps pair



Ode à la joie





Conception et Architecture



Vue d'ensemble

- read_midi/
 - Midi.cpp, Track.cpp, Header.cpp, MidiEvent.cpp, MetaEvent.cpp, Data.h
 - write_wav.cpp
 - write_midi.cpp
 - compare_midi.cpp
- fichiers concernant détection de notes
- yin.cpp
 - monophonic_detection.cpp

Étapes

Étude sur les sons et les formats de fichiers

Conversion midi -> wav

- Parser .mid
- Fichier audio.wav monophonique

Conversion wav -> midi

- Détection note
- Détection mélodie
- Détection onset / offset

Difficultés à la conception

- Connaissances physiques
- Détection de notes



Algorithme

YIN : estimation hauteur d'une note



Programmation : Algorithme YIN

- Diviser le signal audio en fenêtres de taille fixe
- Pour chaque fenêtre appliquer ces étapes
 - Etape 1 : autocorrélation / différence

```
fonction autocorrelation(signal_audio)
    delta = 0
    pour i de 0 à len(signal_audio) :
        pour tau de 0 à len(signal_audio) :
            delta = signal_audio[i] - signal_audio[i + tau]
            buffer[i] = buffer[i] + delta * delta
```

- Etape 2 : moyenne cumulative de la différence

```
fonction cmnd(difference : tableau)
    sum = 0
    buffer[0] = 1
    pour i de 1 à len(difference) :
        sum = sum + buffer[i]
        buffer[i] = buffer[i] * i / sum
```

Programmation : Algorithme YIN

- Etape 3 : Seuil absolu

```
fonction seuilAbsolu(cmnd : tableau)
  i : int # seuil absolu
  pour i de 2 à len(cmnd) :
    si buffer[i] < SEUIL :
      tant que (i+1 < len(cmnd))
        and (buffer[i+1] < buffer[i]) :
          i = i + 1

  si (i == len(cmnd))
  or (buffer[i] >= SEUIL) :
    i = -1
  retourne i
```

- Etape 4 : Interpolation parabolique

```
fonction interpolation_parabole(seuil_est : int) :
  seuil : int
  x1 = seuil_est - 1
  x2 = seuil_est
  x3 = seuil_est + 1

  y1 = buffer[x1]
  y2 = buffer[x2]
  y3 = buffer[x3]

  seuil = seuil_est + (y3 - y1) /
    |(2 * (2 * y2 - y3 - y1))
  retourn seuil
```

```
pitch = sampleRate / interpolation_parabole(seuilAbsolu)
```



Tests effectués



Tests

- Conversion dans les deux sens (wav \rightarrow midi et midi \rightarrow wav)
- Distance de Levenshtein personnalisé sur les MidiEvent
 - Distance sur les notes
 - Distance sur le temps de début de chaque note
 - Distance sur le temps de fin de chaque note

Tests

Fichier gamme do

```
Distance de levenshtein sur les notes 0(/8)
Distance de levenshtein sur temps de début 0(/8)
Distance de levenshtein sur temps de fin 0(/8)
```

Fichier ode à la joie

```
Distance de levenshtein sur les notes 38(/62)
Distance de levenshtein sur temps de début 38(/62)
Distance de levenshtein sur temps de fin 38(/62)
```

Fichier gamme do avec silence

```
Distance de levenshtein sur les notes 0(/4)
Distance de levenshtein sur temps de début 0(/4)
Distance de levenshtein sur temps de fin 0(/4)
```


Conclusion

- Version améliorée :
 - meilleure détection d'onset et d'offset
 - possibilité d'imiter son d'instruments dans la conversion midi à wav
 - tenter détection de mélodie polyphonique
- Si à refaire ...