Reconnaissance des notes de musique

TIPE : François Pech & Armand Gonthier

Juin 2017

Objectifs et motivations

Plan de la présentation

I/ Le son des instruments à l'ordinateur

1. Encodage sur ordinateur - Audacity

II/ Méthode de décomposition

1. Principe de Fourier
2. Méthode
3. Code version α

III/ Interprétation des fréquences

1. Détection d'une note
2. Repérage d'un enchaînement de note

IV/ Short Time Transform Fourier

1. Explication de l'algorithme de la STFT
2. Code
3. Interprétation des données

V/ Ecriture de la partition

1. Tempo et mesure
2. Lilypond

Conclusion

I)
Le son des instruments à l'ordinateur

Encodage sur un ordinateur - Audacity

Micro ⟶ Ordinateur

Format .wav

II) Méthode de décomposition

Principe de Fourier

Transformée de Fourier

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t) \text{ où } \omega = \frac{2\pi}{T}$$

et où $a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos(n\omega t)\,dt$, $b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\sin(n\omega t)\,dt$ et $a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)dt$

$\Longleftrightarrow$

Transformée de Fourier discrète (TFD)

Principe de Fourier

Algorithme de la FFT

$$f(t) = \sum_{n=0}^{N-1} c_n \, e^{in\omega t}$$

Symétrie $$e^{i\omega\left(k+\frac{T}{2}\right)} = -e^{i\omega k}$$

Périodicité $$e^{i\omega(k+T)} = e^{i\omega k}$$

Méthode

- “Diviser pour mieux régner”
- Somme de deux TFDs de plus petite longueur
- Bien plus rapide !

$$\sum_{n=0}^{N-1} c_n \, e^{i\omega n t} = \sum_{r=0}^{\frac{N}{2}-1} c_{2r} e^{i\omega 2rt} + \sum_{r=0}^{\frac{N}{2}-1} c_{2r+1} e^{i\omega(2r+1)t}$$

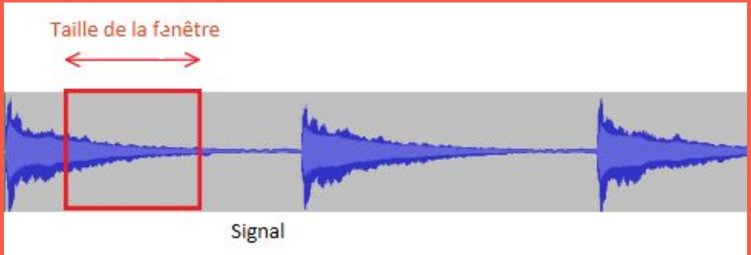
$$e^{i\omega 2rt} = e^{i\omega \frac{2\pi}{N} 2rt} = e^{i\omega \frac{2\pi}{\frac{N}{2}} rt}$$

$$\sum_{n=0}^{N-1} c_n \, e^{i\omega n t} = \sum_{r=0}^{\frac{N}{2}-1} c_{2r} e^{i\omega \frac{2\pi}{\frac{N}{2}} rt} + e^{i\omega t} \sum_{r=0}^{\frac{N}{2}-1} c_{2r+1} e^{i\omega \frac{2\pi}{\frac{N}{2}} rt}$$

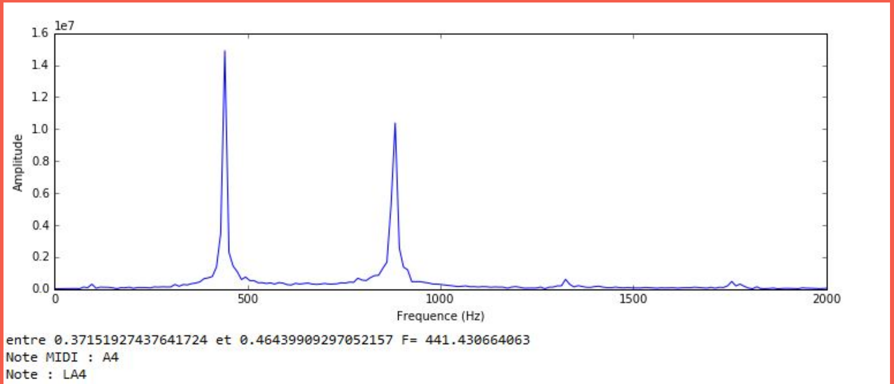
Code version α

```python
def deplacement_fft(i):
    # calcul de la TFD
    s = signal[0+fenetre_FFT*i:fenetre_FFT*(i+1)]
    signal_FFT = abs(fft(s)) # on ne récupère les composantes réelles

    # récupération du domaine fréquentiel en passant la période d'échantillonnage
    Freq_FFT = fftfreq(s.size, dt)

    # extraction des valeurs réelles de la FFT et du domaine fréquentiel#
    signal_FFT = signal_FFT[0:len(signal_FFT)//2]
    Freq_FFT = Freq_FFT[0:len(Freq_FFT)//2]

    #affichage du spectre du signal
    plt.figure(figsize=(12,4))
    plt.xlabel('Frequence (Hz)'); plt.ylabel('Amplitude')
    Freq_min = 0
    Freq_max = 2000
    plt.xlim(Freq_min,Freq_max)
    plt.plot(Freq_FFT,signal_FFT)
    plt.show()
    print('entre',( 0+fenetre_FFT*i)*dt,'et',
          (fenetre_FFT*(i+1))*dt,'F=',Freq_FFT[np.argmax(signal_FFT)])

i=0
while (i+1)*fenetre_FFT < len(signal):  #deplacement de la fenetre fft
    deplacement_fft(i)
    i+=1
```

Fenêtre glissante

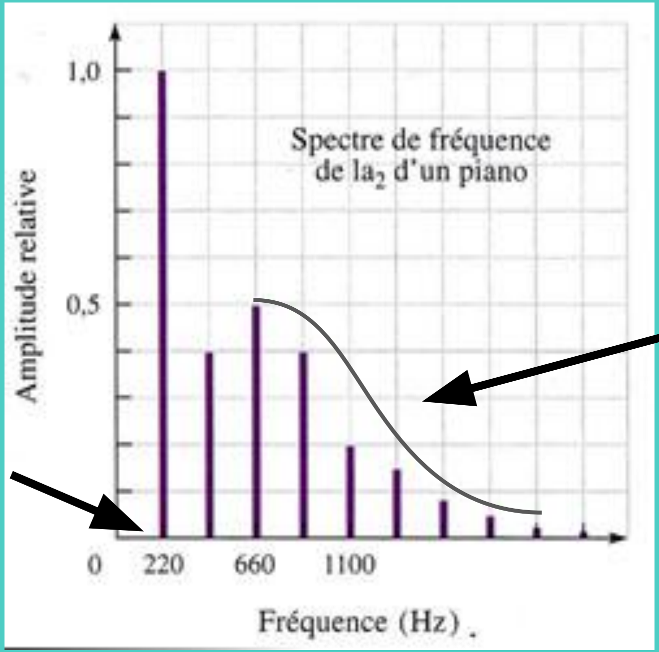


Application de la FFT :



III) Interprétation des fréquences

Caractéristiques des sons

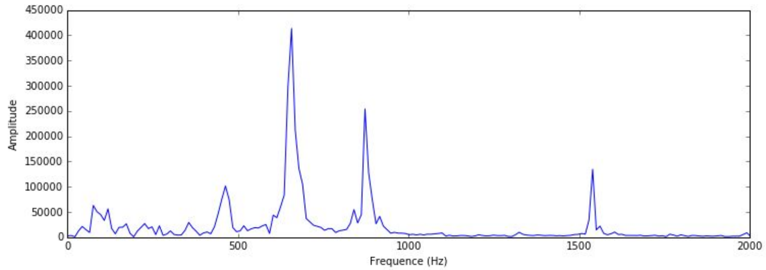


Harmoniques

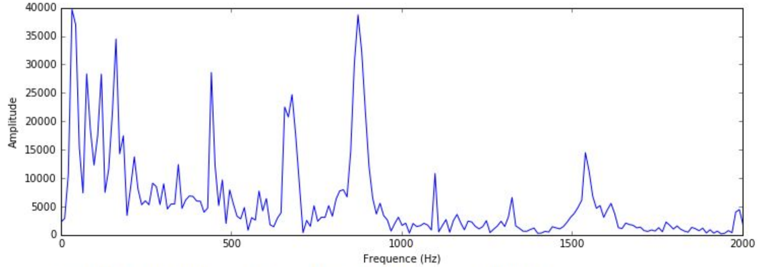
Fréquence fondamentale

Repérage d'un enchaînement de notes

Amplitude



entre 2.8792743764172335 et 2.972154195011338 F= 656.762695313



entre 2.972154195011338 et 3.0650340136054424 F= 32.2998046875

Enchaînement
de fréquences égales

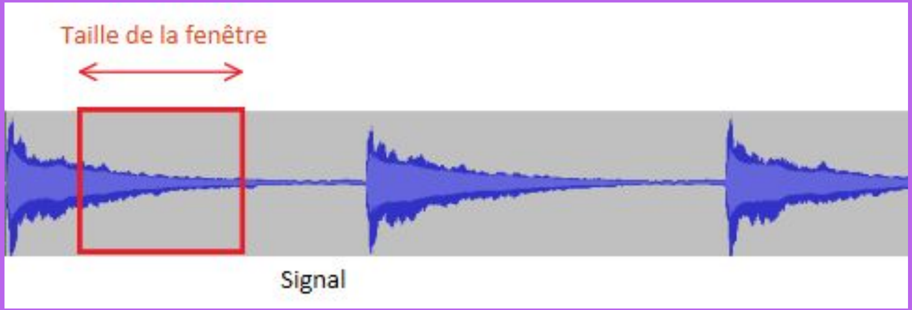
entre 3.0650340136054424 et 3.1579138321995464 F= 785.961914063
entre 3.1579138321995464 et 3.250793650793651 F= 785.961914063
entre 3.250793650793651 et 3.3436734693877552 F= 785.961914063
entre 3.3436734693877552 et 3.4365532879818597 F= 785.961914063
entre 3.4365532879818597 et 3.5294331065759637 F= 785.961914063
entre 3.5294331065759637 et 3.622312925170068 F= 785.961914063
entre 3.622312925170068 et 3.7151927437641725 F= 785.961914063

IV)

Short Time Fourier Transform

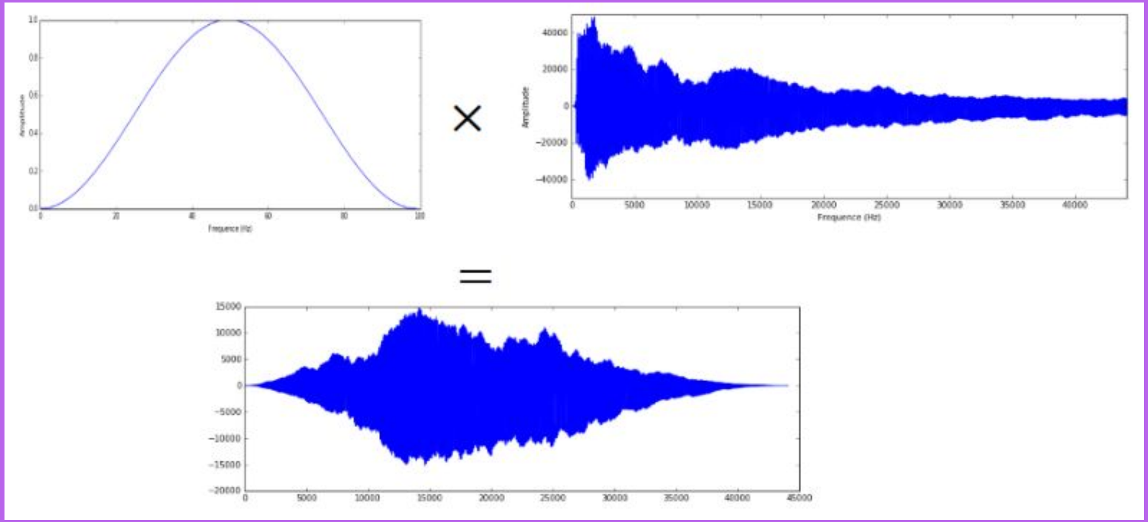
Explication de l'algorithme de la STFT

FFT sur des segments très courts ⇒ Problème ☹



Explication de l’algorithme de la STFT

Multiplication par un demi-cosinus (Fenêtre de Hann)



Code

Agrandissement artificiel de notre signal

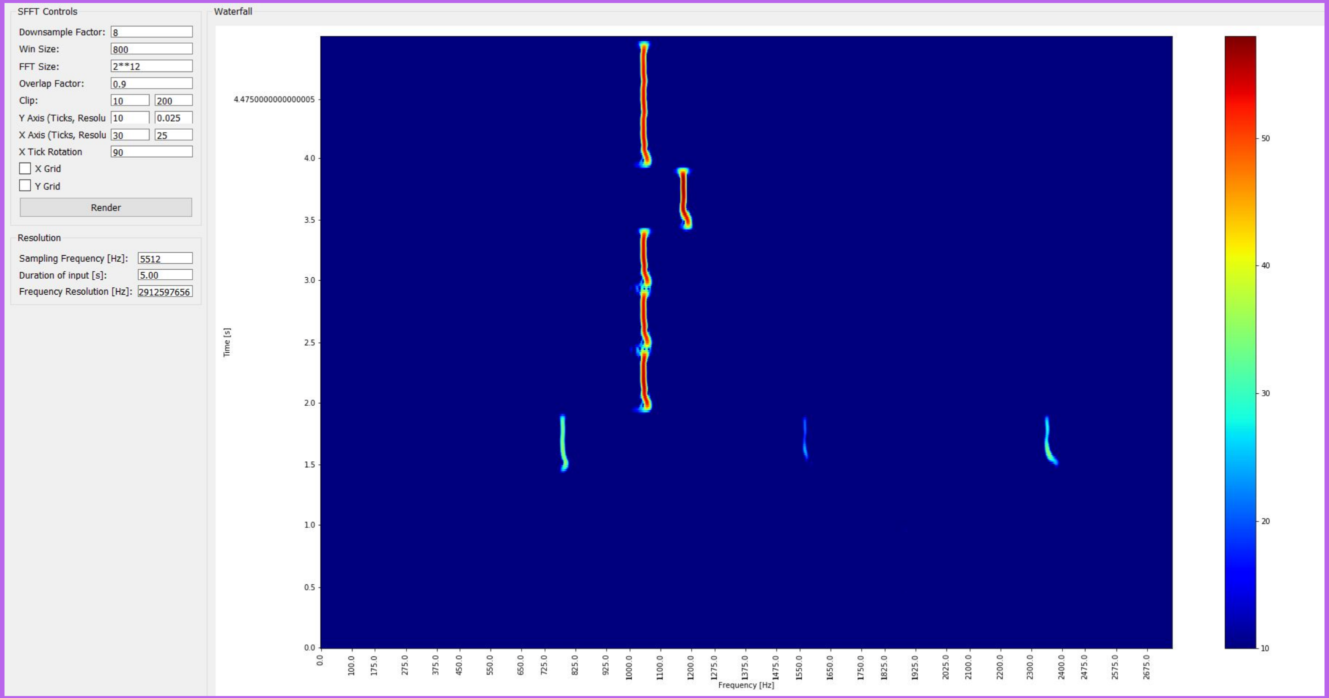
Multiplication par la fenêtre de Hann

Application de la FFT

Extraction des données utiles

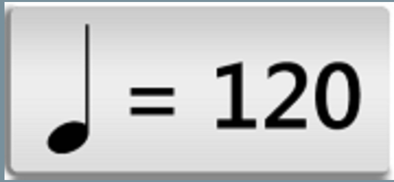
```
ile = 'C:\\Users\\Armand\\Desktop\\TIPE\\piano.wav'
s,data = read(nomfile)
if len(data.shape) != 1:
    data = np.array(data.sum(axis=1), dtype='float64')
fft_size = eval("2**15")

chevauchement = 0.9

reechantillonage = 8

data = sig.decimate(data, reechantillonage, ftype='fir')
fs = fs // reechantillonage

win_size = 1000
taille_saut = np.int32(np.floor(win_size * (1-chevauchement)))
pad_end_size = fft_size
total_segments = np.int32(np.ceil(len(data) / np.float32(taille_saut)))
t_max = len(data) / np.float32(fs)
window = np.hanning(win_size) * chevauchement * 2
inner_pad = np.zeros((fft_size * 2) - win_size)

proc = np.concatenate((data, np.zeros(pad_end_size)))
result = np.empty((total_segments, fft_size), dtype=np.float32)

for i in range(total_segments):
    saut = taille_saut * i
    t = (saut + win_size)/fs
    segment = proc[saut:saut + win_size]
    windowed = segment * window
    padded = np.append(windowed, inner_pad)
    spectrum = np.fft.fft(padded) / fft_size
    autopower = np.abs(spectrum * np.conj(spectrum))
    result[i] = autopower[:fft_size]

    freqSpectrum = np.fft.fftfreq(spectrum.size, 1/fs)
    #print("Entre {}s et {}s, la fréquence max est {}".format(saut/fs, t if t <= t

result = 20*np.log10(result)
result = np.clip(result, -60, 200)
```

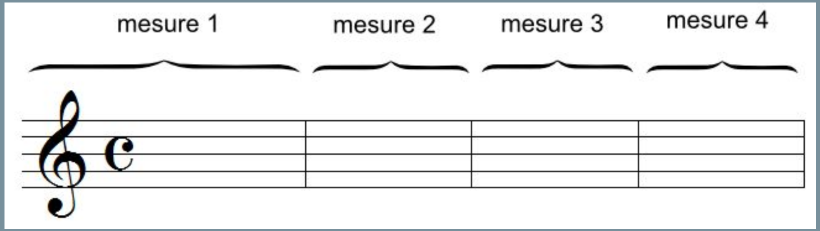
Interprétation des données



IV) Ecriture de la partition

Tempo et mesure






```
def trad_lilypad(time,note,tempo):
    if time < (tempo/60)/3:
        return Note[note]+'16'
    if time < (tempo/60)*3/4:
        return Note[note]+'8'
    if time < (tempo/60)*5/4:
        return Note[note]+'4'
    if time < (tempo/60)*7/4:
        return Note[note]+'4.'
    if time < (tempo/60)*5/2:
        return Note[note]+'2'
    if time < (tempo/60)*7/2:
        return Note[note]+'2.'
    if time < (tempo/60)*4:
        return Note[note]
```

lilypond®

data.ly - LilyPad

Fichier Edition Rechercher Aide

```
\header {
  title = "flute"
  composer = "TIPE"
  tagline = "Copyright: Gonthier, Pech"
}{
\newStaff {
 g'4. a'16 a'16 b''4 a''16 a'8 g'8 b''8 g'4. g'16 b''4 a''16 a'8 g'8
b''8 g'4. g'16 a'16 b''4 a''16 a'8 g'8 b''8 g'4.}
}
```

Conclusion