AGH AGH University of Science and Technology

Python – Projekt 1 – analiza dźwięków

W ramach projektu wykorzystać będzie trzeba dotychczas zdobytą wiedzę oraz umieć zastosować pewne nowe rozwiązania (według wskazówek). Projekt składa się z 2 części. W ramach każdej z nich należy przygotować program w języku Python oraz zbiorczo (dla obu części) krótkie (max. 3-4 strony) sprawozdanie, w którym umieszczone zostaną istotne fragmenty kody wraz z komentarzami, opisane wykresy i wyniki oraz komentarz/wnioski.

Część 1 – analiza pojedynczych tonów (75% punktów)

Należy:

1) Wczytać dane z pliku wav (do tego celu proszę zapoznać się z dokumentacją wavefile.read() z biblioteki scipy.io, na podstawie ilości próbek oraz częstotliwości próbkowania wygenerować wektor czasu z użyciem np.linspace().
2) Dla wskazanego (patrz tabela poniżej) pliku wav z folderu "tony" wygenerować wykres amplitudy od czasu w pełnym zakresie czasu (nazwę pliku zapisać na stałe w programie – nie ma konieczności pobierania jej od użytkownika)

Ostatnia cyfra numeru indeksu	Plik *.wav	Oczekiwana częstotliwość [Hz]
9 lub 0	C4	261.6
1	D4	293.7
2	E4	329.6
3	F4	349.2
4	G4	392.0
5	A4	440.0
6	H4	493.9
7	C5	523.3
8	D5	587.3

3) Wyświetlić wykres modyfikując skalę za pomocą xlim() w taki sposób aby można było odczytać okres przebiegu np. poprzez zliczenie ilości przejść przez zero w określonej jednostce czasu. Po jego odczytaniu z wykresu należy obliczyć częstotliwość (ręcznie)
4) Wykonać fitowanie funkcji typu sinusosida gasnąca do wczytanych danych z pliku, do tego celu należy usunąć z dane przed wzbudzeniem (należy podać w programie ilość próbek z początku przebiegu, które należy usunąć) oraz skorzystać ze wzoru który dopasuje dodatkowo przesunięcie czasowe phi (to nie było uwzględniane w poprzedniej instrukcji na ten temat),

$$y = A \cdot \sin(2 \cdot \pi \cdot f \cdot t + \boldsymbol{\varphi}) \cdot e^{-\gamma t}$$ *, gdzie*:

- $A - amplituda\ oscylacji$
- $f - częstotliwość\ oscylacji$
- $\gamma - współczynnik\ tłumienia > 0,$
- $\varphi - przesunięcie\ czasowe$

5) Określić częstotliwość przebiegu na podstawie dopasowania (fitowania)
6) Wygenerować FFT (Fast Fourier Transform) korzystając z funkcji z poprzedniej instrukcji
7) Odczytać z FFT jaka była częstotliwość oscylacji
8) Porównać wszystkie 3 częstotliwości wyznaczone z powyższych metod

W sprawozdaniu należy umieścić fragmenty kodu programu opatrzone komentarzami, odpowiednio podpisane wykresy (przebiegi czasowe i spektrum z FFT), wzór funkcji fitowania, tabelę porównującą wyniki oraz wnioski.

Część 2 – analiza melodii (25% punktów)

Należy:

1) Wczytać dane z pliku wav (wybrać można jedną z 4 melodii lub, dla większego wyzwania, plik z „akordami" składającymi się z 2 dźwięków każdy)
2) Zapoznać się z działaniem funkcji plt.specgram(data, Fs=samplerate, NFFT=1024), która generuje spektrogram, czyli wykres zmian widma w czasie.
3) Za pomocą plt.ylim() ustawić tak granice, aby obserwować interesujący nas zakres częstotliwości (np. do 1000 Hz)
4) Dobrać wartość parametru NFFT, aby uzyskać dobrą rozdzielczość w osi częstotliwości dla wybranego zakresu.
5) Znając częstotliwości odpowiadające określonym dźwiękom (patrz np. http://www.michalkaszczyszyn.com/pl/lessons/notes.html) zidentyfikuj wszystkie dźwięki występujące w melodii.

W sprawozdaniu zamieść spektrogram z ręcznie (np. Paint) naniesionymi adnotacjami dot. rozpoznanych dźwięków.
Jeśli grasz na jakimś instrumencie możesz spróbować przeanalizować w ten sposób własne nagranie. Czy wiesz z jakiego instrumentu pochodzą dostarczone nagrania?

Opracowali: dr inż. Sławomir Ziętek, mgr inż. Piotr Rzeszut, Oprawa artystyczna: mgr inż. Piotr Rzeszut