Wydział Mechatroniki Politechniki Warszawskiej

Warszawa



**Zespolona Analiza Fouriera z Wizualizacją**

Jednostka modułowa:

ZAP II – Zasady programowania strukturalnego

Wykonał:

Tomasz Tober

04 marzec 2025

​​**Spis treści**

[Założenia programu 4](#_u1lckrg4spqs)

[Wewnętrzne działanie Projektu 4](#_u1lckrg4spqs)

[Instrukcja obsługi etapu pierwszego 5](#_u1lckrg4spqs)

# 

​

​

​

​​

### **Założenia programu**

projekt jest prawie w całości edukacyjny i ezoteryczny, nie ma pełnić żadnej funkcji w prawdziwym życiu. Ma on jednak spełniać pewne założenia, ma pozwalać na swobodne wprowadzenie przez użytkownika krzywej, ma on następnie odpowiednio efektywnie przetworzyć krzywą zachowując charakterystyczne aspekty krzywej, a następnie sporządzić, zapisać i zwizualizować transform krzywej na zbiór obracających się ramion.

To są jedynie podstawowe założenia przewidziane na pierwszy etap.

### **Wewnętrzne działanie Projektu**

Rola i zastosowanie każdego z plików:

program ze względów technicznych wykorzystuje bibliotekę graficzną QT, oraz wewnętrzne moduły do komunikacji z biblioteką graficzną (displaty.h/.cpp, VectorDisplay.h/.cpp), do przechowywania i przetwarzania danych(Path.h/cpp, Complex.h/cpp)

Wewnętrzna ścieżka danych(jak są przetwarzane dane):

tworzone są dwie struktury dynamiczna tablica pokazująca punkty(tylko do wizualizacji) , oraz struktura Shape zawierająca połączona lista dwukierunkowa typu Path nazwana current, następnie display.h wraz z biblioteką QT, tworzą i wyświetlają w formie bitmapy okienko pokazujące tablice. Używając Eventów przechwytywane są ruchy i kliknięcia myszką, gdy klawisz myszki jest wciśnięty i porusza się, dodawane są nowe elementy do ścieżki, oraz zmieniany jest kolor w tablicy.   
w momencie ponownego naciśnięcia klawisza całość listy jest wymazywana.

W momencie naciśnięcia klawisza Enter, krzywa zostaje poddana obróbce:  
  
1.najpierw jest na niej wykonywany algorytm RDP (Rammer-Douglas-Peucker), który usuwa kolejne punkty jeżeli są bliskie (0.5px) od współliniowych z już istniejącym, usuwa to bardzo dużą ilość punktów które nijak nie dodawały do zrozumienia krzywej, jednocześnie zostawiając te które są potrzebne do około dokładnego odtworzenia jej.  
 z wizualnych powodów następnie updatuje bitmapę pokazując połączone te punkty, pokazując to że prawie żadne detale nie zostają stracone w procesie, a niektóre artefakty, wejścia danych zostaną poprawione(losowe przerwy wynikające z prędkości poruszania się myszki).

2. następnie z otrzymanych charakterystycznych punktów interpoluje się odpowiednią ilość punktów pośrednich, aby uzyskać liczbę punktów jako jakąś potęgę dwójki.

robi to w taki sposób że najpierw liczy ile punktów trzeba by było dodać do obecnego stanu żeby osiągnąć taki stan, a następnie sporządza listę najdłuższych połączeń trzymając ich dystans, adres jednego z punktów i liczbę planowanych cięć(default 1). następnie znajduje optymalny sposób na rozdystrybuowanie tych cięć, tworzy dwa “pointery” na początku i końcu listy i sprawdzając czy odcinki powstałe przez stworzenie dodatkowego punktu w najdłuższym odcinku, nadal byłyby większe od najmniejszego elementu listy, i jeżeli tak , wyrzuca najmniejszy element i dodaje dodatkowe cięcie do najdłuższego, jeżeli nie, to zeszło by o element niżej, aż oba pointery się spotkają.  
następnie dodawane są zaplanowane punkty.  
O(n)

3.Ostatnią operacją na tych danych jest faktyczny transfor, fouriera, ja zdecydowałem się wykorzystać szybki transform fouriera(FFT, Cooley-Turkey), który wykorzystuje pewne powtarzające się wartości w toku sinusów i cosinusów(w moim przypadku to e^ift), żeby osiągnąć O(n log(n)) zamiast standardowego n^2 dla zwykłego DFT(discrete fourier transform).

Przed poddaniem tych punktów algorytmowi, zamieniam je na liczby zespolone.

Po wszystkich tych operacjach jedyne rzeczy jakie nam pozostały to zapis i wizualizacja.

Zapis w obecnej wersji jest wyjątkowo uproszczony, zapisując współczynniki w pliku tekstowym wewnątrz plików programu. pod nazwą FC.txt

a wizualizacja wywołuje nowy process, wyświetlający obracające się ramiona. Dodatkowo dla ułatwia zobaczenia faktycznego efektu, dodałem smuge punktów poprzednich pozycji końcówki ramienia.

Plany na kolejny etap:  
-umożliwianie wczytywania i bardziej precyzyjnego zapisywania krzywych

-możliwość wprowadzenia obrazów z plików i automatyczne generowanie z nich krzywych

- możliwa jest szansa zmiany typu algorytmu wykorzystanego do transformu by uzyskać łagodniejsze zmiany(wektoryzacja i B-SPL FT)

### **Instrukcja obsługi etapu pierwszego**

po włączeniu programu pojawi się białe okno, użytkownik może na nim rysować myszką, w momencie naciśnięcia klawisza Enter, wszystkie poza ostatnią ciągłą krzywą narysowaną zniknie. Krzywa zostanie nieznacznie zarysowana, oraz otworzy się nowe okno pokazujące obracające się ramiona wyznaczające tą krzywą. Dodatkowo współczynniki potrzebne do zrekonstruowania tych ramion są zawsze odświeżane w pliku tekstowym FC.txt w miejscu wywołania pliku.