

Zadanie B: Transformata Fouriera (20 punktów)

Napisz program który dla danego rysunku będzie liczyć dyskretną transformatę Fouriera i jej odwrotność.

Zakładamy że obraz wejściowy:

- Jest obrazem kwadratowym
- Wymiar boku jest wyrażony liczbą 2^k , $k \in \mathbb{N}$
- Wszystkie piksele są szare (składowe R, G i B są sobie równe)

Transformata Fouriera powinna wygenerować obraz tej samej wielkości, gdzie składowa R reprezentuje część rzeczywistą transformaty, zaś G – część urojona. B powinno pozostać równe 0. Powstała transformata nie powinna być przesuwana, t.j. wartość $F(0,0)$ powinna być umieszczona w górnym lewym pikselu, a nie na przykład w środku rysunku.

Zarówno transformata jak i jej odwrotność powinny mieć taki sam czynnik normalizujący $\sqrt{\frac{1}{MN}}$, gdzie M times N to rozmiar obrazu.

Szczegóły zadania

Powyższe zadanie należy rozwiązać w ramach projektu nad którym pracujemy na ćwiczeniach.

Program powinien obsługiwać komendę: `fft <in> <out>` który policzy transformatę dla rysunku `<in>` a wynik umieści w rysunku `<out>`.

Analogicznie, komenda: `ifft <in> <out>` policzy transformatę odwrotną, tworząc biało-czarny rysunek `<out>`. Złożenie `fft` i `ifft` musi dać dokładnie ten sam obraz co w oryginale.

Uwaga! Nazwy `<in>` i `<out>` to nie nazwy plików, tylko wewnętrzne nazwy w projekcie. Zapisywanie na dysk odbywa się tylko za pomocą komendy `save`.

Argument do `ifft` może być rysunkiem który nie mógł powstać w wyniku transformaty z rysunku rzeczywistego. W takim wypadku odwrotna transformata może zawierać komponenty urojone. Należy je zignorować, t.j. przy rekonstrukcji bierzemy jedynie część rzeczywistą.

Program powinien też obsługiwać komendy:

- `exit`
- `load`
- `save`
- `get`
- `create`
- `fill`
- `put`

Komendy te były wyjaśnione na ćwiczeniach. Argumenty które są kolorami powinny akceptować parametry R G B podane bezpośrednio, bez specyfikacji przestrzeni barw.

Przykład

Dla następującego wejścia:

```
create R 4 4
fill R 0.5 0.5 0.5
put R 0 0 0 0 0
put R 1 0 0 0 0
put R 2 0 0 0 0
put R 3 0 0 0 0
put R 0 2 1 1 1
put R 1 2 1 1 1
put R 2 2 1 1 1
fft R F
get F 0 0
get F 1 0
get F 2 0
get F 3 0
get F 0 1
get F 0 2
get F 0 3
get F 1 1
get F 2 2
get F 3 3
ifft F R2
get R2 0 0
get R2 1 1
get R2 2 2
get R2 3 3
get R2 3 2
exit 0
```

Program powinien wypisać na standardowe wyjście:

```
Program GK
R: 1.875 G: 0 B: 0
R: 0 G: -0.125 B: 0
R: 0.125 G: 0 B: 0
R: 0 G: 0.125 B: 0
R: -0.875 G: 0 B: 0
R: -0.125 G: 0 B: 0
R: -0.875 G: 0 B: 0
R: 0 G: 0.125 B: 0
R: 0.125 G: 0 B: 0
R: 0 G: -0.125 B: 0
R: 0 G: 0 B: 0
R: 0.5 G: 0.5 B: 0.5
R: 1 G: 1 B: 1
R: 0.5 G: 0.5 B: 0.5
R: 0.5 G: 0.5 B: 0.5
Exiting with code 0
```

Wartości liczb rzeczywistych mogą się nieznacznie różnić. Dopuszczalny błąd to 0.001.