**Opis programu**

Program służy do filtracji obrazów trójwymiarowych przy użyciu filtrów: medianowego, uśredniającego, wyostrzającego oraz filtru Sobla, który wykrywa krawędzie we wszystkich kierunkach. Analizie możemy poddać obraz zapisany w formie tekstowej lub binarnej. Wykorzystywane są w nim moduły tworzone przy użyciu Cythona, dzięki czemu otrzymujemy czasy działania porównywalne z programami napisanymi w języku C lub C++. Do wczytywania oraz zapisu wykorzystywane są funkcje z biblioteki NumPy.

**Instalacji i kompilacja**

W celu uruchomienia należy posiadać kompilator C++ oraz środowisko Pythona. Następnie zainstalować biblioteki Cython oraz NumPy. Można to łatwo zrobić używając pip komendą:

***pip install Cython/NumPy***

Po zainstalowaniu można przystąpić do tworzenia modułów. W tym celu wykonujemy komendę:

***python setup.py build\_ext --inplace***

Wykonując tą komendę, z plików zawierających implementację filtrów w języku c++, tworzymy, z użyciem plików Cythona, nowy plik c++ tzw. plik modułowy. Następnie plik ten jest kompilowany do modułu Pythona dołączanego do programu. Pliki Cythona pozwalają na opakowanie filtrów do postaci funkcji, które mogą zostać wykorzystane w Pythonie.

Po stworzeniu modułów program uruchamiamy komendą

***python Filtry.py <znak filtru> <nazwa pliku>***

Znak filtru określa, który z filtr chcemy zastosować na danych zawartych w pliku. Odpowiednie znaki odpowiadają:

* -m filtr medianowy
* -u filtr uśredniający
* -m filtr medianowy
* -s filtr Sobla

Nazwa pliku musi być zapisana w formacie:

***<rozmiarX>x<rozmiarY>x<rozmiarZ>.<rozszerzenie>***

Gdzie rozmiarX, rozmiarY, rozmiarZ określają wielkość analizowanego obrazu w kierunkach X,Y,Z zaś rozszerzenie mówi jaki rodzaj danych zawiera plik. Pliki binarne rozpoznawane przez program posiadają rozszerzenie bin oraz raw. Każdy inny jest traktowany jako plik tesktowy.

**Algorytmy**

W programie wykorzystano algorytmy zgodne z następującymi pseudokodami:

* Filtr medianowy

Stwórz NowyObraz

Stwórz Tablice

Margines = (maska-1)/2

For i in każda warstwa

For j in każdy wiersz

For k in każdy element

Ilosc = 0

For x in <-margines,margines>

For y in <-margines,margines>

For z in <-margines,margines>

Tablica[ilosc] = Obraz[i+x,j+y,k+z]

Ilsoc++

Posortuj(Tablica)

NowyObraz[i,j,k] = mediana(Tablica)

Wyczyść(Tablica)

Zwróc NowyObraz

* Filtr uśredniający

Stwórz NowyObraz

Margines = (maska-1)/2

Size = pow(maska,3)

For i in każda warstwa

For j in każdy wiersz

For k in każdy element

Suma = 0

For x in <-margines,margines>

For y in <-margines,margines>

For z in <-margines,margines>

Suma += Obraz[i+x,j+y,k+z]

NowyObraz[i,j,k] = Suma/Size

Zwróc NowyObraz

* Filtr wyostrzający

Stwórz NowyObraz

Margines = (maska-1)/2

Size = pow(maska,3)

For i in każda warstwa

For j in każdy wiersz

For k in każdy element

Suma = 0

For x in <-margines,margines>

For y in <-margines,margines>

For z in <-margines,margines>

If x==y==z==0 Suma += 27\*Obraz[i+x,j+y,k+z]

Else Suma += -Obraz[i+x,j+y,k+z]

NowyObraz[i,j,k] = Suma/Size

Zwróc NowyObraz

* Filtr Sobla – wartość piksela określamy według wzoru



Sx, Sy, Sz to sumy wartości po nałożeniu masek kierunkowych, przedstawiających się nastepująco:

* Kierunek Z

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -1 | -3 | -1 |  | 0 | 0 | 0 |  | 1 | 3 | 1 |
| -3 | -6 | -3 |  | 0 | 0 | 0 |  | 3 | 6 | 3 |
| -1 | -3 | -1 |  | 0 | 0 | 0 |  | 1 | 3 | 1 |

* Kierunek X

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 1 |  | 3 | 6 | 3 |  | 1 | 3 | 1 |
| 0 | 0 | 0 |  | 0 | 0 | 0 |  | 0 | 0 | 0 |
| -1 | -3 | -1 |  | -3 | -6 | -3 |  | -1 | -3 | -1 |

* Kierunek Y

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -1 | 0 | 1 |  | -3 | 0 | 3 |  | -1 | 0 | 1 |
| -3 | 0 | 3 |  | -6 | 0 | 6 |  | -3 | 0 | 3 |
| -1 | 0 | 1 |  | -3 | 0 | 3 |  | -1 | 0 | 1 |