

Projekt 1

Brytan Filip

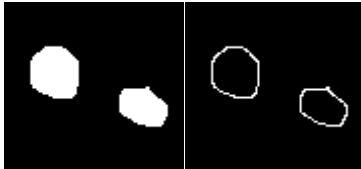
298236

1.Program

Program został przeze mnie napisany w języku C++ w Visual Studio z użyciem bibliotek System i System::Drawing. W przypadku gdy funkcje miały działać dla dwóch typów obrazów, program sam rozpoznaje czy ma do czynienia z obrazem RGB czy monochromem.

2.Wypełnianie dziur w obiektach

Działanie funkcji testowałem na obrazku wykonanym przez siebie



Jak i również z Pana strony:





Wyniki są identyczne jak w matlabie.

Algorytm wygląda następująco:

- 1) Tworzę 3 obrazy, jeden który jest zanegowanym wejściowym, drugi wynikowym, trzeci który na początku jest 1px ramką o wartości 1
- 2) Wykonuje dylację geodezyjną elementem w kształcie plusa
- 3) Porównuję ją z zanegowanym obrazem wejściowym
- 4) Wykonuje to dopóki dylacja nie zmieni obrazu
- 5) Po przerwaniu pętli, neguję otrzymany wynik i otrzymuję wypełniony obraz który zapisuje

3. Otwarcie elementem liniowym o zadanej długości i nachyleniu

Funkcję ponownie testuje na obrazie, z linią o nachyleniu 60stopni i długości 10



Po wykonaniu wygląda on tak:



Jest tutaj jednak lekka różnica względem matlaba, ale o tym niżej.

Z kolei dla monochromu testuje dla cameramana:



Ponownie jest jedna różnica względem matlaba. Wynika ona z następującej rzeczy:

Matlab rysuje linię w lekko inny sposób tzn:

| | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 | 0 |
| 7 | 0 | 1 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 |
| 10 | | | | | |

Podczas gdy algorytm Bresenhama z którego korzystałem (https://pl.wikipedia.org/wiki/Algorytm_Bresenhama)

Rysuje ją w ten sposób:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

Niemniej jednak otwarcie jest wykonywane w sposób prawidłowy, tylko na delikatnie innej linii.

Algorytm wygląda następująco:

- 1) Po podaniu długości i nachylenia liczę położenie punktów:

```
ptk_1 = (cos( radian ) * length);
ptk_2 = (sin( radian ) * length);
```

- 2) Rysuję do tablicy linię korzystając z algorytmu Bresenhama
- 3) Wykonuję najpierw dylację, później erozję. Pamiętając o przesunięciu punktu do środka linii
- 4) Wynik zapisuje

4. Ordfilt2

Algorytm z maską 3x3 i numerem porządkowym 9 wykonuję na cameramanie:



Wynik jest identyczny jak w matlabie.

Dla lamy:

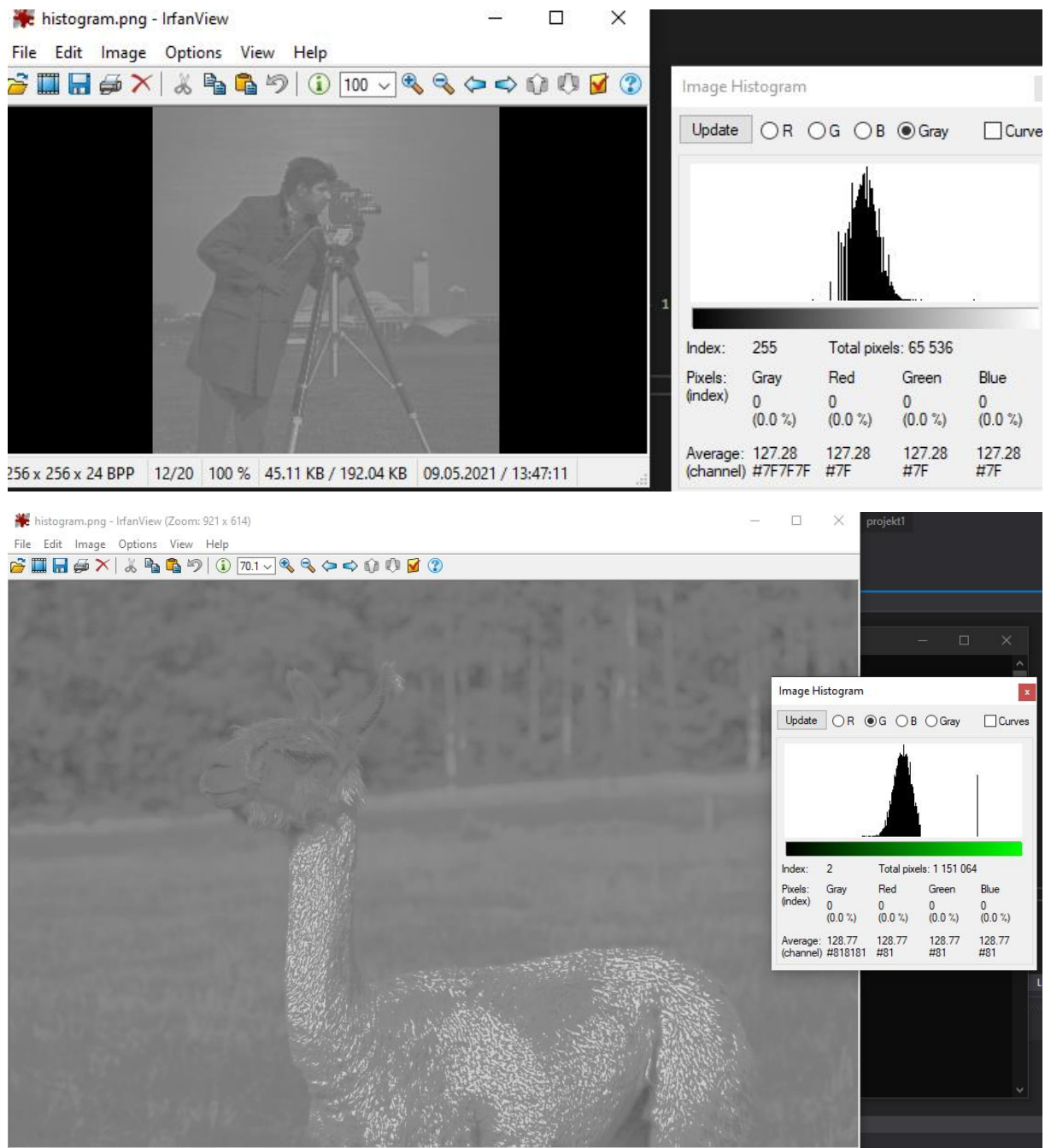


Algorytm:

- 1) Tworzę obraz wyjściowy o rozmiarze takim jak wejściowy
- 2) Tworzę maskę $N \times M$
- 3) Zbieram wartości pixeli w odległości od maski dla danej pozycji do tablicy wynikowej (dla rogów i wartości skrajnych nie biorę pod uwagę pozycji wychodzących za obraz)

- 4) Sortuje tablicę wynikową i ustawiam wartość pixela w danym miejscu w obrazie wyjściowym `wyn[numer_porzadkowy]`
- 5) Zapisuję obraz wynikowy

5. Wyrównanie histogramu



Algorytm:

- 1) Pobieram wartości pixeli z obrazka i tworzę histogram (dla rgb każda warstwa osobno)

- 2) Liczę dystrybuantę: `histGr[i] += histGr[i - 1];`
- 3) Później liczę histogram ze wzoru: `exp(-1 * pow(i - srednia, 2) / (2. * pow(odch, 2))) / (odch * sqrt(2. * M_PI));`
- 4) Następnie dystrybuantę, którą później wyrównuję + wymnażam * szerokość i wysokość obrazka
- 5) Później porównuję dystrybuantę z obrazka z gaussowską i tworzę LUT
- 6) Wypełniam obraz wartościami z LUT. (Dla RGB mam 3 tablice z LUT)