### Informatyka, studia dzienne, mgr II st. semestr I

## Przetwarzanie obrazu i dźwięku 2010/2011

### Prowadzący: dr inż. Bartłomiej Stasiak poniedziałek, 12:15

### Data oddania: Ocena:

Aleksander Dominiak 234049

Dominik Andrzejczak 234038

# Zadanie 1:

# Szkielet aplikacji do przetwarzania i analizy obrazów, operacje podstawowe, usuwanie szumu, modyfikacje histogramu, filtracja liniowa i nieliniowa, splot.

## 1. Cel

Celem niniejszego zadania było zaprojektowanie szkieletu aplikacji do przetwarzania i analizy obrazów. Dodatkowo należało zaimplementować podstawowe operacje takie jak: zmiana kontrastu czy jasności, wyznaczenie negatywu obrazu. Innymi funkcjonalnościami programu miały byś filtr uśredniający i medianowy, a także przekształcenia histogramu, filtry liniowe oparte o splot oraz filtry nieliniowe.

## 2. Wprowadzenie

Wykonanie zadania wiązało się z implementacją operacji przetwarzania obrazu, które oparte są na konkretnej wiedzy i algorytmach. Każda z nich wykorzystuje piksele, z których złożony jest obraz, do jego modyfikacji. W celu realizacji pierwszego zadania zbudowane aplikację, która umożliwia użytkownikowi zastosowanie poniższych filtrów i modyfikacji obrazu:

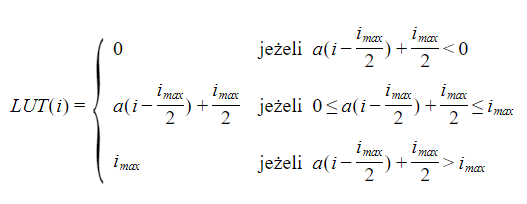
1. Zmiana jasności
2. Zmiana kontrastu
3. Wyznaczenie negatywu
4. Filtr średni arytmetyczny
5. Filtr medianowy
6. Filtr liniowy (północ, północny - wschód, wschód, południowy – wschód)
7. Filtr nieliniowy – Operator Rosenfelda
8. Uzyskanie oczekiwanej charakterystyki histogramu obrazu wynikowego

Zmiana jasności obrazu polega na dodaniu do każdej wartości piksela określonej wartości przez użytkownika. Jeżeli chcemy zwiększyć lub zminiejszyć jasność obrazu o x to należy dla każdego piksela wykonać działanie dodawania lub odejmowania:

Pi,j = Pi,j + x lub Pi,j = Pi,j - x

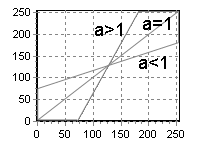
Gdzie, Pi,j – piksel i-tego wiersza i j-kolumny obrazu

Modyfikując kontrast obrazu warto zastosować LUT (Look Up Table) [1]. Jest to tabela przechowująca zdefiniowane nowe wartości pikseli. Przygotowanie tablicy LUT opiera się na rozwiązaniu równania liniowego ax + b :



gdzie,imax – maksymalna możliwa wartość piksela (255)

W przypadku standardowych wartości pikseli maksymalną wartością jest 255. Tak więc utworzenie tabeli LUT sprowadza się do rozwiązania równania a(i – 127) + 127 dla wszystkich możliwych wartości piksela i z przedziału [0, 255]. Parametr a jest modyfikowany przez użytkownika co prowadzi do zmian wykresu powyższej funkcji:

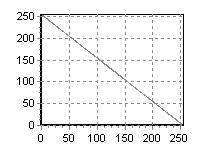


Gdy mamy już zdefiniowany LUT możemy modyfikować piksele na obrazie przypisując im odpowiednie wartości. Dokonujemy przypisania dla każdego piksela obrazu oryginalnego tak, że i = LUT[i].

Podobną metodę przyjęto na potrzeby wyznaczenia negatywu obrazu. Wykorzystanie tablicy LUT pozwala na zaoszczędzenie zasobów podczas obliczeń dla każdego piksela. W tym przypadku powstawanie tablicy LUT opiera się na odjęciu od możliwej maksymalnej wartości - i-tego piksela [2]:



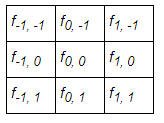
Co odpowiada wykresowi funkcji:



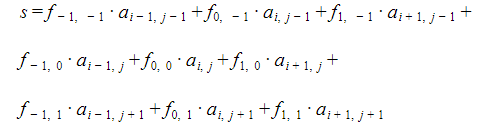
Filtr uśredniający to wykorzystanie tzw. maski o określonym rozmiarze (typowe maski są o rozmiarach 3x3, 5x5 lub 7x7) do skorygowania szumu na obrazie. Przez ich zastosowanie obraz zostaje rozmyty. Maska filtru uśredniającego 3x3 przedstawia poniższa tabela:

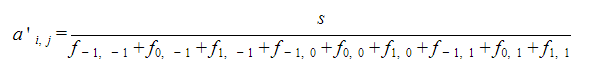
|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Maskę stosujemy dla każdego piksela z wyjątkiem pierwszych i ostatnich wierszy i kolumn. Jest to jeden ze sposobów uniknięcia problemu pikseli granicznych, dla których zastosowanie maski powoduje wykroczenie poza obraz. Następnie przyjmując ai,j jako aktualną wartość modyfikowanego piksela możemy zmienić ją na średnią wartość jego sąsiadów. Oznaczając kolejne pola maski jako:



Możemy zastosować wzory stosowane do stosowania różnych filtrów:





Gdzie, a’i,j to nowa wartość piksela (średnia sąsiednich pikseli i piksela badanego)

Z uwagi na postać maski otrzymujemy średnią wartość wszystkich sąsiednich pikseli oraz piksela, który jest modyfikowany. Liczba pikseli sąsiadujących jest uzależniona od wielkości maski.

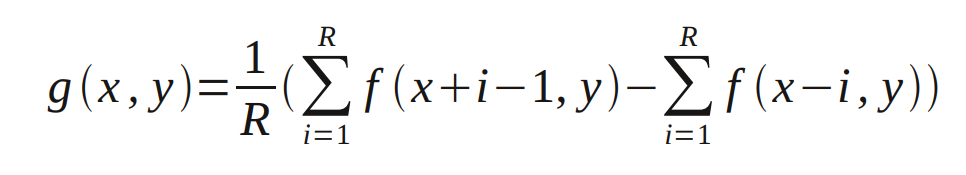
Dla filtru medianowego również możemy zdefiniować taką maskę. Jednak późniejsze obliczenia nie są potrzebne. W przypadku tej operacji należy posortować wszystkie piksele sąsiadujące malejąco lub rosnąco w zależności od ich wartości. Następnie wybieramy wartość środkową, czyli zwyczajną medianę i przypisujemy ją jako nową wartość piksela.

Filtry liniowe północ, północny-wschód, wschód i południowy-wschód służą między innymi do wyznaczania krawędzi na obrazie. Każdemu z powyższych odpowiada maska:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | północ | | | | 1 | 1 | 1 | | 1 | -2 | 1 | | -1 | -1 | -1 | | |  |  |  | | --- | --- | --- | | pół.-wschód  chód | | | | 1 | 1 | 1 | | -1 | -2 | 1 | | -1 | -1 | 1 | | |  |  |  | | --- | --- | --- | | wschód | | | | -1 | 1 | 1 | | -1 | -2 | 1 | | -1 | 1 | 1 | | |  |  |  | | --- | --- | --- | | poł.-wschód | | | | -1 | -1 | 1 | | -1 | -2 | 1 | | 1 | 1 | 1 | |

Następnie wartość piksela wyznaczana jest zgodnie z metodą przedstawioną dla filtrów liniowych. Z zastrzeżeniem, że suma wartości pól w masce jest równa 0, więc nie możemy podzielić sumy przez sumę pól maski. Dlatego gdy wartość obliczonej sumy s > 255 to nowa wartość piksela (a’i,j) = 255, a gdy s < 0 to (a’i,j) = 0.

Filtr nieliniowy w postaci operatora Rosenfelda opiera się o równanie:

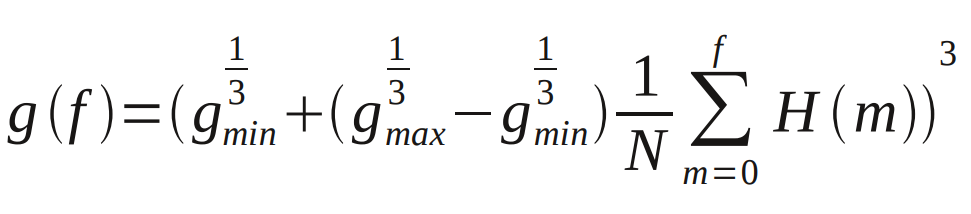


gdzie R jest wartością przekazaną przez użytkownika.

Wartość piksela wyznaczona za pomocą wzoru powyżej wynika z wartości wybranej liczby pikseli znajdujących się w tym samym rzędzie. Jest to różnica sumy danego piksela i *R-1* następnych pikseli i sumy poprzednich pikseli podzielona przez *R*.

Histogram przedstawia rozkład empiryczny cech, w danym wypadku tą cechą jest jasność dla obrazów czarno białych lub wartości poszczególnych kanałów dla kolorowych. Można go przedstawić w postaci wykresu np. jasności od częstotliwości występowania.

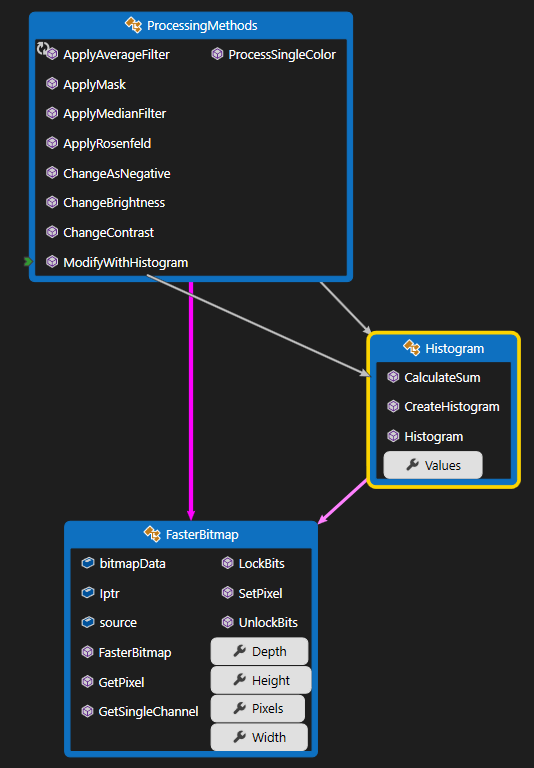
Histogram może zostać wykorzystany do zmiany obrazu, np. w oparciu o wzor



gdzie *H* to histogram obrazu źródłowego, *N* liczba pikseli w obrazie, a *gmin* i *gmax* są wartościami pobranymi od użytkownika. W ten sposób można zmienić charakterystykę histogramu obrazu wynikowego tak, by uzyskane wartości na osi *x* (jasność, wartości kanałów) znajdowały się w wybranym przedziale.

## 3. Opis implementacji

Aplikacja została napisana w języku C# z użyciem frameworka Windows Forma.



Klasa FasterBitmap odpowiada za przyśpieszenie edycji wbudowanej klasy bitmapy. Pozwala ona na pobranie i ustawienie wartości danego piksela lub w przypadku obrazów kolorowych konkretnego kanału.

Klasa Histogram oblicza i przechowuje wartości potrzebne do stworzenia wykresu.

Statyczna klasa ProcessingMethods to zbiór metod modyfikujących bitmapę w oparciu o przekazane wartości.

## 4. Materiały i metody

Z wykorzystaniem stworzonej aplikacji przeprowadzono badania dla wybranych obrazów (a-h) w celu zweryfikowania poprawności ich działania. Zastosowano wszystkie zaimplementowane operacje i filtry do wszystkich poniższych wariantów:

|  |  |
| --- | --- |
|  |  |
| a | b |
|  |  |
| c | d |
|  |  |
| e | f |
|  |  |
| g | h |

W tym miejscu należy opisać, jak przeprowadzone zostały wszystkie badania, których wyniki i dyskusja zamieszczane są w dalszych sekcjach. Opis ten powinien być na tyle dokładny, aby osoba czytająca go potrafiła wszystkie przeprowadzone badania samodzielnie powtórzyć w celu zweryfikowania ich poprawności (a zatem m.in. należy zamieścić tu opis architektury sieci, wartości współczynników użytych w kolejnych eksperymentach, sposób inicjalizacji wag, metodę uczenia itp. oraz informacje o danych, na których prowadzone były badania). Przy opisie należy odwoływać się i stosować do opisanych w sekcji drugiej wzorów i oznaczeń, a także w jasny sposób opisać cel konkretnego testu. Najlepiej byłoby wyraźnie wyszczególnić (ponumerować) poszczególne eksperymenty tak, aby łatwo było się do nich odwoływać dalej.

## 5. Wyniki

W tej sekcji należy zaprezentować, dla każdego przeprowadzonego eksperymentu, kompletny zestaw wyników w postaci tabel, wykresów itp. Powinny być one tak ponazywane, aby było wiadomo, do czego się odnoszą. Wszystkie tabele i wykresy należy oczywiście opisać (opisać co jest na osiach, w kolumnach itd.) stosując się do przyjętych wcześniej oznaczeń. Nie należy tu komentować i interpretować wyników, gdyż miejsce na to jest w kolejnej sekcji. Tu również dobrze jest wprowadzić oznaczenia (tabel, wykresów) aby móc się do nich odwoływać poniżej.

## 6. Dyskusja

Sekcja ta powinna zawierać dokładną interpretację uzyskanych wyników eksperymentów wraz ze szczegółowymi wnioskami z nich płynącymi. Najcenniejsze są, rzecz jasna, wnioski o charakterze uniwersalnym, które mogą być istotne przy innych, podobnych zadaniach. Należy również omówić i wyjaśnić wszystkie napotakane problemy (jeśli takie były). Każdy wniosek powinien mieć poparcie we wcześniej przeprowadzonych eksperymentach ( odwołania do konkretnych wyników). Jest to jedna z najważniejszych sekcji tego sprawozdania, gdyż prezentuje poziom zrozumienia badanego problemu.

## 7. Wnioski

W tej, przedostatniej, sekcji należy zamieścić podsumowanie najważniejszych wniosków z sekcji poprzedniej. Najlepiej jest je po prostu wypunktować. Znów, tak jak poprzednio, najistotniejsze są wnioski o charakterze uniwersalnym.

## Literatura

[1] <http://www.algorytm.org/przetwarzanie-obrazow/zmiana-kontrastu-obrazu.html>

[2] <http://www.algorytm.org/przetwarzanie-obrazow/negatyw-obrazu.html>