

Przetwarzanie wstępne (*preprocessing*). Filtracja kontekstowa.

Cel:

- zapoznanie z pojęciem kontekstu / filtracji kontekstowej
- zapoznanie z pojęciem konwolucji (spłotu)
- zapoznanie z wybranymi metodami poprawy jakości obrazu:
 - filtry liniowe dolnoprzepustowe (uśredniające):
 - uśrednianie
 - Gauss
 - filtry nieliniowe
 - mediana
 - mediana dla obrazów kolorowych
 - filtry liniowe górnoprzepustowe (wystraszające)
 - laplajsan
 - operator Robersta, Prewitta, Sobela,
- zapoznanie z adaptacyjnym filtrem medianowym

Przekształcenia kontekstowe:

Przekształcenia kontekstowe są to przekształcenia przy których dla wyznaczenia wartości jednego punktu obrazu wynikowego trzeba dokonać określonych obliczeń na wielu punktach obrazu źródłowego (znajdujących się w otoczeniu rozważanego punktu).

A. Filtry liniowe uśredniające (dolnoprzepustowe)

Jest to najprostsza rodzina filtrów stosowana w przetwarzaniu obrazów. Wykorzystuje się je w celu roznazania obrazu i redukcji szumów (zakłóceń) na obrazie. Filtr określony jest przez dwa parametry: rozmiar maski (ang. *kernel*) oraz wartości współczynników maski.

1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na swój własny katalog na dysku D. Otwórz nowy m-plik (**New Script**) lub (**New->Script**). Na początku wykonaj polecenia `close all; clearvars; clc;`
2. Wczytaj obraz "plansza.tif" (wcześniej ściągając archiwum ze strony www.iro.paukuj.com w odpowiednim katalogu). W dalszej części ćwiczenia sprawdzenie działania filtracji dla innych obrazów sprowadzi się do wczytania innego pliku.
3. Podstawowa funkcja to `conv2` - realizacja konwolucji. Zwróć uwagę na parametr 'shape'. Ma on związek z tzw. problemem brzegowym (na krawędziach istnieją piksele dla których nie da się wyznaczyć otoczenia). Dla potrzeb tego laboratorium dobrze jest zachować rozmiar
4. Stwórz podstawowy filtr uśredniający o rozmiarze 3x3 - za pomocą funkcji `fspecial` z parametrem 'average'. Wykonaj konwolucję na wczytanym obrazie. Wynik przekonwertuj do formatu `uint8`. Na wspólnym rysunku wyświetl obraz oryginalny, po filtracji oraz modul z różnicy:
 - wykorzystaj polecenie `subplot`
 - do obliczania modułu z różnicy przydatna jest funkcja `imabsdiff`
 - do wyświetlania różnicy wykorzystaj następującą postać: `imshow(...,[]);`
5. Przeanalizuj otrzymany wynik. Jakiego elementu zawiera obraz "moduł z różnicy"? Co na tej podstawie można powiedzieć o filtracji dolnoprzepustowej?
6. Na wspólnym rysunku wyświetl wyniki filtracji uśredniającej z oknem o rozmiarze 3, 5, 9, 15 i 35. Wykorzystaj polecenie `subplot`. Przeanalizuj wpływ rozmiaru maski na wynik. Drobna uwaga - podczas pracy starajmy się nie usuwać napisanego kodu - lepiej jest go zakomentować, tak aby możliwe było jego ponowne wykorzystanie.
7. Wczytaj obraz "lena.bmp". Zaobserwuj efekty filtracji dolnoprzepustowej dla obrazu rzeczywistego.
8. Niekorzystny efekt towarzyszący wykonanym filtracjom dolnoprzepustowym to utrata ostrości. Częściowo można go zniwelować poprzez dobór maski.
$$M = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Wprowadź maskę do Matlab. ($M = [1 \ 2 \ 1; \ 2 \ 4 \ 2; \ 1 \ 2 \ 1];$). Przed obliczeniami należy jeszcze wykonać normalizację - podzielić każdy element maski przez sumę wszystkich elementów: $M = M / \text{sum}(\text{sum}(M))$; Tak przygotowaną maskę wykorzystaj w konwolucji - wyświetl wyniki tak jak w punkcie 4. Możliwe jest też wykorzystywanie innych masek - współczynniki dostosowuje się do konkretnego problemu.
9. Skuteczną i często wykorzystywaną maskę jest tzw. maska Gaussa. Jest to zbiór liczb, które aproksymują dwuwymiarowy rozkład Gaussa. Parametrem jest odchylenie standardowe.
10. Wykorzystując funkcję `fspecial` z parametrem 'gaussian' stwórz maskę o rozmiarze 5x5 i odchyleniu standardowym 0.5. Wykorzystaj polecenie `mesh` zwiualizuj filtr. Sprawdź jak parametr „odchylenie standardowe” wpływa na „kształt” filtru.
11. Wykonaj filtrację dla wybranych (2-3) wartości odchylenia standardowego.
12. [P] Zaprezentuj wyniki pracy prowadzącemu.

B. Filtry nieliniowe - mediana

Filtry rozmywające redukują szum ale niekorzystnie wpływają na ostrość obrazu. Dlatego często wykorzystuje się filtry nieliniowe - np. filtr medianowy (mediana - średnia wartość w posortowanym ciągu liczb).

Podstawowa różnica pomiędzy filtrami liniowymi, a nieliniowymi polega na tym, że przy filtracji liniowej na nową wartość piksela ma wpływ wartość wszystkich pikseli z otoczenia (uśrednianie), natomiast w przypadku filtracji nieliniowej jako nowy piksel wybierana jest któraś z wartości otoczenia - według jakiegoś wskaźnika (wartość największa, najmniejsza czy właśnie mediana).

1. Otwórz nowy m-plik. Na początku wykonaj polecenia `close all; clearvars; clc;` Wczytaj obraz "lenaSzum.bmp" (losowe 10% pikseli białych lub czarnych - tzw. zakłócenia impulsowe). Wykorzystując funkcję `medfilt2` wykonaj filtrację medianową (z domyślnym rozmiarem maski 3x3). Wyświetl, podobnie jak wcześniej, oryginalny, wynik filtracji i modul z różnicy. Odpowiedz - czy filtracja medianowa dobrze usuwa zakłócenia impulsowe (przypadkowe powiedz o wartości 0 lub 255)?
2. Przeprowadź filtrację medianową, a następnie uśredniającą obrazu "lena.bmp". Wyniki porównaj - dla obu wyświetl: oryginalny, wynik filtracji i modul z różnicy. Szczególną uwagę zwróć na ostrość i krawędzie. Kolejna ważna cecha filtracji medianowej - położenie krawędzi zostaje zachowane.
3. Ciekawy efekt można uzyskać wykonując filtrację medianową wielokrotnie. Określa się go mianem posteryzacji. W wyniku przetwarzania z obrazka usunięte zostają detale, a duże obszary uzyskują tą samą wartość jasności. Wykonaj operację mediany 5x5 na obrazie "lena.bmp" 10-krotnie. (wykorzystaj np. pętlę `for`)
4. [P] Zaprezentuj wyniki pracy prowadzącemu.
5. Idea filtracji medianowej jest dość prosta dla obrazów w odcieniach szarości. Dla obrazów kolorowych trudniej jest określić kryterium wg. którego szeregowane będą wartości, z których wyznaczana będzie mediana.
6. Jedną z możliwości wykonania filtracji medianowej dla obrazów kolorowych (na podstawie The Image Processing Handbook, J. Russ) jest wykorzystanie następującej definicji mediany:
 - mediana to ten piksel z otoczenia, którego odległość do innych pikseli z otoczenia jest najmniejsza
 - jako miarę odległości wykorzystujemy pierwiastek z sumy kwadratów różnic poszczególnych składowych R,G,B - zatem odległość między dwoma pikselami wyraża się wzorem:
$$o = \sqrt{((R1-R2)^2 + (G1-G2)^2 + (B1-B2)^2)}$$

Warto zwrócić uwagę, że istnieje wiele możliwości zdefiniowania porównywania wielkości wektorowych (jeden piksel to wektor trzech składowych (RGB, HSV)). Można zamiast odległości wykorzystać kąt albo połączyć oba parametry. Ponadto istnieje możliwość dodania do wektora dodatkowych składowych - tak aby lepiej opisać piksel.

7. Przykładowa implementacja (demonstracyjna) opisanej metody zawarta jest w pliku "median3dRGB.m". Uruchom skrypt (uwaga - obliczenia trwają chwilę). zapoznaj się z wynikami. Zmień obraz na "lenaRGBSzum.bmp". Ponownie uruchom skrypt.

Inne filtry nieliniowe:

- filtr modyowy - moda (dominanta) zamiast mediany
- filtr olimpijski - średnia z podzbioru otoczenia (bez wartości ekstremalnych)
- hybrydowy filtr medianowy - mediana obliczana osobno w różnych podzbiorach otoczenia ("x", "u"), a jako wynik brana jest mediana ze zbioru wartości elementu centralnego, mediana z "x" i mediana z "u"
- filtr minimalny i maksymalny (będą omówione przy okazji operacji morfologicznych)

Warto zdawać sobie sprawę, że szerokiś możliwości dopasowywania rodzaju filtracji do konkretnego rozważanego problemu i rodzaju zażamienia rodzaju elementów faktury.

C. Filtry liniowe górnoprzepustowe (wystraszające, wykrywające krawędzie)

Zadaniem filtrów górnoprzepustowych jest wydobywanie z obrazu składników odpowiedzialnych za szybkie zmiany jasności - konturów, krawędzi, drobnych elementów faktury.

Laplajsan (wykorzystanie drugiej pochodnej obrazu)

1. Otwórz nowy m-plik. Na początku wykonaj polecenia `close all; clearvars; clc;` Wczytaj obraz "moon.tif".
2. Wprowadź podstawową maskę laplajsanu:
$$M = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
3. Przed rozpoczęciem obliczeń należy dokonać normalizacji maski - dla rozmiaru 3x3 podzielić każdy element przez 9.
4. Wykonaj konwolucję obrazu z maską (`conv2`). Pamiętaj o parametrze 'same'. Przed wyświetleniem, wynikowy obraz należy poddać normalizacji (ujemne wartości). Najczęściej wykonuje się jedną z dwóch operacji:
 - skalowanie (np. poprzez dodatnie 128 do każdego z pikseli)
 - modul (wartość bezwzględna)Wykonaj obie normalizacje. Na wspólnym wykresie wyświetl obraz oryginalny oraz przefiltrowany po obu normalizacjach. Uwaga: przy wyświetlaniu wykorzystaj opcję skalowania zakresu: `imshow(obraz,[],)`

5. Do tworzenia masek laplajsanu można wykorzystać polecenie `fspecial` z parametrem 'laplacian'. Zapoznaj się z dokumentacją (szczególnie z algorytmem wyznaczania współczynników). Wykonaj filtrację z maską stworzoną za pomocą funkcji `fspecial`.

6. Efekt wyostrzenia uzyskuje się po odjęciu/dodaniu (zależy do maski) rezultatu filtracji laplajsanowej (przed normalizacją) i oryginalnego obrazu. Do wykresu dodaj czwarty obraz: różnicę/sumę oryginalu i filtracji (może zająć konieczność konwersji typów). Wykorzystaj opcję skalowania zakresu.
7. [P] Wyniki zaprezentuj prowadzącemu.

Gradienty (wykorzystanie pierwszej pochodnej obrazu)

1. Otwórz nowy m-plik. Na początku wykonaj polecenia `close all; clearvars; clc;` Wczytaj obraz "kw.bmp". Wczytaj maski za pomocą polecenia `load mask1PP`. Wykorzystując gradient Roberta przeprowadź detekcję krawędzi - poprzez wykonanie konwolucji obrazu z daną maską:
$$\begin{matrix} -0 & -0 & 0 & & -0 & -0 & 0 \\ R1 = & -1 & -0 & -0 & & R2 = & -0 & -0 & -1 \\ & -0 & +1 & -0 & & & -0 & +1 & -0 \end{matrix}$$
Wykorzystaj stworzony wcześniej kod (przy laplajsanie) - dwie metody normalizacji oraz sposób wyświetlania.

2. Analogicznie przeprowadź detekcję krawędzi z pomocą gradientu Prewitta (pionowy i poziomy)
$$\begin{matrix} & -1 & -0 & +1 & & & -1 & -1 & -1 \\ P1 = & -1 & -0 & +1 & & P2 = & -0 & -0 & -0 \\ & -1 & -0 & +1 & & & +1 & +1 & +1 \end{matrix}$$

3. Podobnie skonstruowany jest gradient Sobela (występuje osiem masek, zaprezentowane są dwie)
$$\begin{matrix} & -1 & -0 & +1 & & & -1 & -2 & -1 \\ S1 = & -2 & -0 & +2 & & S2 = & -0 & -0 & -0 \\ & -1 & -0 & +1 & & & +1 & +2 & +1 \end{matrix}$$

Przeprowadź detekcję krawędzi z pomocą gradientu Sobela.

4. Na podstawie dwóch ortogonalnych masek Sobela można stworzyć tzw. filtr kombinowany - pierwiastek kwadratowy z sumy kwadratów gradientów: $OW = \sqrt{((O * S1)^2 + (O * S2)^2)}$ OW - obraz wyjściowy, O - obraz oryginalny, S1,S2 - maski Sobela, * - operacja konwolucji. Zaimplementuj filtr kombinowany (Uwaga - przy podnoszeniu wyniku do kwadratu należy użyć operatora '^2' - zapewni to podniesienie każdego z elementów macierzy do kwadratu).

5. Istnieje alternatywna wersja filtra kombinowanego, która zamiast pierwiastka z sumy kwadratów wykorzystuje sumę modułów (prostsze obliczenia). Zaimplementuj tę wersję. Obraz oryginalny oraz wyniki filtracji przedstaw na wspólnym wykresie.

6. Wczytaj plik "jet.bmp" (zamiast "kw.bmp"). Sprawdź działanie filtracji.

7. [P] Wyniki zaprezentuj prowadzącemu.

Zadanie dodatkowe***:

- implementacja adaptacyjnego filtra medianowego (na podstawie **Digital Image Processing, Rafael C Gonzalez, Richard E. Woods**).

Wszystkie omówione i przetestowane powyżej filtry działały w ten sam sposób dla każdego piksela obrazu. Sposób działania filtrów adaptacyjnych zmienia się w zależności od statystycznych charakterystyk otoczenia rozpatrywanego piksela.

Adaptacyjny filtr medianowy.

Omówiony w rozdziale B filtr medianowy dobrze eliminuje zakłócenia impulsowe o niewielkiej gęstości przestrzennej (mniej niż 20% w pionie i poziomie). Do filtracji bardziej zakłóconych obrazów dobrze nadaje się adaptacyjny filtr medianowy.

"Adaptacyjność" filtra polega na zmianie rozmiaru okna w trakcie filtracji - w zależności od pewnych warunków.

Jeżeli przyjmijmy oznaczenia:

- Z_{min} - najmniejsza jasność w oknie S_{xy}
- Z_{max} - największa jasność w oknie S_{xy}
- Z_{med} - mediana z jasności w oknie S_{xy}
- Z_{xy} - jasność w punkcie o współrzędnych (x,y)
- S_{max} - maksymalny dozwolony rozmiar okna S_{xy}

Algorytm można opisać w sposób następujący:

Faza A:

$$\begin{matrix} A1 = Z_{med} - Z_{min} \\ A2 = Z_{med} - Z_{max} \\ \text{JEŻELI } A1 > 0 \vee A2 < 0 \\ \text{to przejdź od fazy B} \\ \text{WPW} \\ \text{zwiększ rozmiar okna} \\ \text{JEŻELI rozmiar okna} \leq S_{max} \\ \text{to powtórz fazę A} \\ \text{WPW} \\ \text{zwróć } Z_{med} \end{matrix}$$

Faza B:

$$\begin{matrix} B1 = Z_{xy} - Z_{min} \\ B2 = Z_{xy} - Z_{max} \\ \text{JEŻELI } B1 > 0 \vee B2 < 0 \\ \text{to powtórz } Z_{xy} \\ \text{WPW} \\ \text{zwróć } Z_{med} \end{matrix}$$

Wyjaśnienie działania:

Cały czas pamiętać, jaki są cele działania filtru:

- usunięcie zakłóceń typu pieprz i sól (*salt and pepper noise*)
- wygładzenie innego rodzaju zakłóceń
- redukcja zniekształceń (pogrubianie albo pocienianie krawędzi)

Wartości Z_{min} i Z_{max} uważa się za zakłócenia impulsowe w danym oknie, nawet jeżeli nie są to największe i najmniejsze wartości w całym obrazie.

Celem fazy A jest określenie czy rezultat filtracji medianowej Z_{med} jest zakłóceniem impulsowym czy nie. Jeżeli spełniona jest nierówność $Z_{min} < Z_{med} < Z_{max}$ wartość Z_{med} nie może być zakłóceniem impulsowym. W takim przypadku przechodzimy do fazy B i sprawdzamy czy piksel Z_{xy} jest zakłóceniem impulsowym. Jeżeli spełniona jest nierówność $Z_{min} < Z_{xy} < Z_{max}$, Z_{xy} nie może być zakłóceniem impulsowym. W takim przypadku algorytm zwraca niezmienną wartość Z_{xy} - w ten sposób zmniejsza się zniekształcenia. Jeżeli nierówność $Z_{min} < Z_{xy} < Z_{max}$ nie jest spełniona oznacza to, że albo $Z_{xy} = Z_{min}$ albo $Z_{xy} = Z_{max}$. W obu przypadkach uznaje się, że wartość Z_{xy} jest zakłóceniem i jako wynik filtracji zwraca się wartość Z_{med} , o której na podstawie fazy A wiadomo, że nie jest zakłóceniem.

Przyjmijmy, że nie została spełniona nierówność $Z_{min} < Z_{med} < Z_{max}$, algorytm wtedy zwiększa rozmiar okna i powtórnie wykonuje fazę A. Dzieje się tak aż do momentu, kiedy algorytm znajdzie medianę, która nie jest zakłóceniem impulsowym albo osiągnięty zostanie maksymalny rozmiar okna. W przypadku gdy osiągnięty zostanie maksymalny rozmiar okna algorytm zwraca wartość Z_{med} . Nie ma gwarancji, że wartość ta nie jest zakłóceniem impulsowym, jednakże odpowiedni dobór maksymalnego rozmiaru okna pozwala zminimalizować prawdopodobieństwo wystąpienia takiej sytuacji.

Po wyznaczeniu wartości po filtracji dla danego piksela, okno filtru się przemieszcza, a algorytm uruchamiany jest dla nowej lokalizacji z parametrami początkowymi.

Zadanie - zaimplementować metodę:

1. Otwórz nowy m-plik. Na początku wykonaj polecenia `close all; clearvars; clc;` Wczytaj obrazy "plykaSzumImp.tif". Wyświetl go.
2. Przefiltruj obraz filtrem medianowym o rozmiarze okna 7x7. Wynik filtracji wyświetl.
3. Zaimplementuj opisany adaptacyjny filtr medianowy. Wskazówki:
 - przed rozpoczęciem obliczeń należy skopiować oryginalny obraz - do tej kopii będą zapisywane wyniki filtracji
 - należy przyjąć maksymalny rozmiar okna na 7x7, a początkowy na 3x3. Zakładamy, że rozmiar filtru powinien być nieparzysty. Dla celów implementacji wygodnie jest przyjąć, że filtr 3x3 opisuje liczba 1, 5x5 - 3 itp.
 - obliczenia trzeba wykonać w pętli `for` (odwiedzić, każdy piksel) - dla ułatwienia zaleca się pominięcie pikseli brzegowych (jeżeli maksymalny rozmiar okna wynosi 7x7 to nie należy dokonywać obliczeń dla 3 pikseli brzegowych)
 - w każdej iteracji należy:
 - przywrócić początkowy rozmiar okna (3x3)
 - "wyciągnąć" otoczenie - do nowej zmiennej przekopiować odpowiedni fragment obrazka (parametry opisujący rozmiar otoczenia należy "uzmiennić"), aby uniknąć pomyłek należy skonstruować okno do formatu *double*

4. [P]. Porównaj rezultaty filtracji medianowej i adaptacyjnej. Wyniki zaprezentuj prowadzącemu.