

Diagnostyka obrazowa

Ćwiczenie piąte Filtrowanie obrazu

1 Cel ćwiczenia

Ćwiczenie ma na celu zapoznanie uczestników kursu „Diagnostyka obrazowa” z pojęciami szumu na obrazie oraz metodami redukcji szumów przez zastosowanie filtrów dostępnych w programie ImageJ.

2 Idea filtracji obrazu

Intuicyjnie, filtracją obrazu możemy nazwać usunięcie z niego szumów (pewnych zakłóceń). W ujęciu matematycznym filtr jest operacją przekształcającą w pewien szczególny sposób jeden obraz w drugi, piksel po pikselu, z uwzględnieniem sąsiedztwa analizowanego punktu. Zatem w odróżnieniu od operacji punktowych, które nie biorą pod uwagę sąsiedztwa punktu, filtry możemy zaliczyć do *operacji kontekstowych*, ponieważ biorą one pod uwagę pewien *kontekst* towarzyszący wybranemu punktowi, czyli jego otoczenie. Analogicznie do operacji morfologicznych omawianych na poprzednich zajęciach, w przypadku filtrów również wykorzystujemy pewien element strukturalny, który tym razem nazywamy *maską*. W przeciwieństwie do przekształceń morfologicznych, filtracja wykonywana jest zawsze, a nie przy spełnionych określonych z góry warunkach.

Do podstawowych zadań filtrów, oprócz usuwania szumów należą:

- rekonstrukcja uszkodzonych fragmentów obrazu,
- poprawa jakości obrazów (braku ostrości, słabego kontrastu),
- wzmocnienie w obrazie pewnych elementów (zgodnie z wybranym wzorcem),
- rozmycie obrazu,
- wydobycie informacji o położeniu krawędzi, rogów obiektów.

Filtry możemy podzielić na: liniowe i nieliniowe. Filtry liniowe wykonują operację wg. pewnej liniowej kombinacji pikseli obrazu wejściowego, natomiast filtry nieliniowe bazują na nieliniowej funkcji pikseli obrazu wejściowego. W obrębie tych dwóch kategorii możemy wyróżnić jeszcze następujące rodzaje filtrów:

- Filtry liniowe
 - uśredniające

- wykrywające krawędzie
- Filtry nieliniowe
 - medianowe
 - adaptacyjne

W kolejnych rozdziałach omówimy bardziej szczegółowo czym charakteryzują się przedstawione rodzaje filtrów.

2.1 Pojęcie szumu

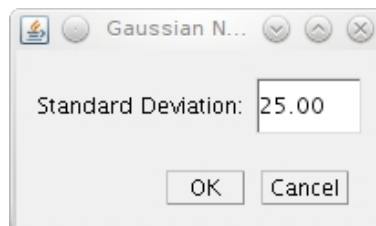
Ponieważ filtracja jest w ogólnym rozumieniu procesem eliminacji szumów z obrazu, należy zatem zrozumieć czym jest samo pojęcie szumu na obrazie. Szum to pewne odchylenie sygnału od jego wartości rzeczywistej. W sposób matematyczny obraz zaszumiony możemy zapisać jako sumę obrazu oryginalnego i jego szumu:

$$g(x, y) = f(x, y) + \mu(x, y),$$

gdzie $g(x, y)$ to piksel o współrzędnych (x, y) na obrazie zaszumionym, f to obraz oryginalny, a μ to szum. Szumy na obrazach są często wynikiem działania aparatury elektronicznej wewnątrz aparatu fotograficznego lub innego urządzenia rejestrującego (kamery, skanera, itp.). Taki rodzaj szumu jest nazywany *ziarnem*. Określenie to oznacza, że obraz składa się z mniej lub bardziej widocznych kropek, które zmniejszają możliwość rozróżnienia szczegółów na obrazie.

Czasami jednak szumy są dodawane celowo do obrazu, w momencie jego tworzenia, na przykład w celu ukrycia nagłego przejścia pomiędzy barwami (tzw. *banding*). Tak zwane *szumy Perlina* są dodawane do obrazu w celu zwiększenia realizmu wizualnego grafiki komputerowej.

W *ImageJ* również mamy możliwość dodania do obrazu różnego rodzaju szumów. W menu Process->Noise znajdują się polecenia pozwalające na dodanie do obrazu szumów. Za pomocą polecenia Add Noise możemy dodać losowy szum do obrazu lub jego wybranego fragmentu. Szum jest rozmieszczany na obrazie zgodnie z rozkładem normalnym o wartości oczekiwanej 0 i standardowym odchyleniu wynoszącym 25. Jeżeli chcemy dodać do obrazu szum o wybranym przez nas odchyleniu standardowym, należy w tym samym menu wykorzystać polecenie Add Specified Noise (**Rysunek 1**) i w jego oknie wybrać odchylenie standardowe dla rozkładu normalnego wg. którego będą generowane szumy.



Rysunek 1: Okno wyboru odchylenia standardowego dla losowego szumu.

2.1.1 Szum typu sól i pieprz

Jednym z najbardziej popularnych rodzajów szumu występujących na obrazach, obok szumu o rozkładzie normalnym, jest szum typu *sól i pieprz*. Z jego nazwy można wywnioskować, że jest on związany z występującymi losowo na obrazie białymi i czarnymi pikselami. W ImageJ można również dodać do obrazu szum typu sól i pieprz za pomocą polecenia `Salt and Pepper` w menu `Process->Noise`. Należy jednak pamiętać, że ten rodzaj szumu możemy dodać tylko do obrazów w skali szarości (8-bitowych). W ImageJ szum typu sól i pieprz losowo zastępuje 2.5% pikseli pikselami czarnymi i odpowiednio 2.5% pikseli pikselami białymi.

2.2 Pojęcie maski filtru

Dla filtrów, element strukturalny wykorzystany w operacji filtracji jest nazywany *maską*. Maską filtru to macierz, najczęściej kwadratowa. Rozmiary masek są często wyrażone za pomocą liczb nieparzystych tj. 3×3 , 5×5 , 7×7 , ponieważ w takim przedstawieniu maski piksel środkowy jest tym, dla którego jest przeprowadzana operacja filtracji. Należy ostrożnie dobierać rozmiary masek, ponieważ im większa maska, tym większa złożoność algorytmu filtracji i większe zmiany obrazu (czasem niepożądane). Wynik działania filtru wynika z przedstawienia maski, która jest w realizacji danego filtru wykorzystywana. Poniższa definicja maski jest przede wszystkim aktualna dla filtrów liniowych. W przypadku filtrów nieliniowych, maska może stanowić po prostu element strukturalny, wraz z pewną regułą, która powinna być wobec niego zastosowana. Nowa wartość każdego z pikseli jest wyliczana za pomocą funkcji, która pobiera wartości pikseli z obrazu obrazu źródłowego (w obrębie maski) i wartości maski, a następnie przeprowadza transformację (np. liczy średnią arytmetyczną, ważoną, itd.).

Przeanalizujemy proces filtracji liniowej na podstawie maski o wymiarze 3×3 przedstawionej przez [Rysunek 2](#), który reprezentuje wagi pikseli, dla pewnego filtra f .

$$\begin{bmatrix} f_{-1,-1} & f_{0,-1} & f_{1,-1} \\ f_{-1,0} & f_{0,0} & f_{1,0} \\ f_{-1,1} & f_{0,1} & f_{1,1} \end{bmatrix}$$

Rysunek 2: Zapis ogólny macierzy wag filtru.

Macierz filtru ([Rysunek 2](#)) jest parowana z pewnym pikselem na obrazie źródłowym ($s_{i,j}$) i jego otoczeniem (patrz: [Rysunek 3](#)).

Wówczas nową wartość składowej $s_{i,j}$ obliczamy w następujący sposób: najpierw obliczymy sumę ważoną punktu (s_w) i jego sąsiadów, zgodnie z wagami wskazywanymi przez maskę filtra (wzór 1).

$$\begin{aligned} s_w = & s_{i-1,j-1} \cdot f_{-1,-1} + s_{i,j-1} \cdot f_{0,-1} + s_{i+1,j-1} \cdot f_{1,-1} + \\ & + s_{i-1,j} \cdot f_{-1,0} + s_{i,j} \cdot f_{0,0} + s_{i+1,j} \cdot f_{1,0} + \\ & + s_{i-1,j+1} \cdot f_{-1,1} + s_{i,j+1} \cdot f_{0,1} + s_{i+1,j+1} \cdot f_{1,1} \end{aligned} \quad (1)$$

$$\begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & s_{2,3} & \cdots & s_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & s_{i-1,j-1} & s_{i,j-1} & s_{i+1,j-1} & \vdots \\ \vdots & s_{i-1,j} & s_{i,j} & s_{j+1,j} & \vdots \\ \vdots & s_{i-1,j+1} & s_{i,j+1} & s_{i+1,j+1} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{m,1} & s_{m,2} & s_{m,3} & \cdots & s_{m,n} \end{bmatrix}$$

Rysunek 3: Zapis ogólny macierzy obrazu.

Następnie, tak otrzymaną sumę dzielimy przez sumę wszystkich wag maski (wzór 2), jeżeli jest ona różna od 0. Ten proces nazywamy normalizacją wartości punktu i zapobiega on np. zmianie jasności przetwarzanego obrazu, w stosunku do obrazu wyjściowego. Przez $s'_{i,j}$ rozumiemy piksel obrazu po zastosowaniu filtracji.

$$s'_{i,j} = s_w / (f_{-1,-1} + f_{0,-1} + f_{1,-1} + f_{-1,0} + f_{0,0} + f_{1,0} + f_{-1,1} + f_{0,1} + f_{1,1}) \quad (2)$$

W ujęciu matematycznym proces filtracji rozumiemy jako działanie nazywane *splotem* macierzy obrazu z maską filtru. Tak samo jak w przypadku przekształceń morfologicznych, należy zdefiniować co dzieje się, kiedy maska dla punktów skrajnych wychodzi poza obszar krawędzi obrazu. Jest kilka sposobów na radzenie sobie z tym problemem:

- zmniejszyć obraz o punkty, dla których proces filtracji był niemożliwy,
- pominąć proces filtracji dla takich punktów,
- dodanie do filtrowanego obrazu zduplikowanych pikseli z jego brzegu.

Jeżeli obraz jest zapisany w trybie RGB, to tego typu proces przeprowadzamy dla każdej składowej obrazu z osobna. Na rysunku poniżej (Rysunek 4) znajdują się znane maski (macierze wag) wykorzystywane do algorytmów filtracji:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Rysunek 4: Przykłady masek o rozmiarach 3x3.

3 Filtry liniowe

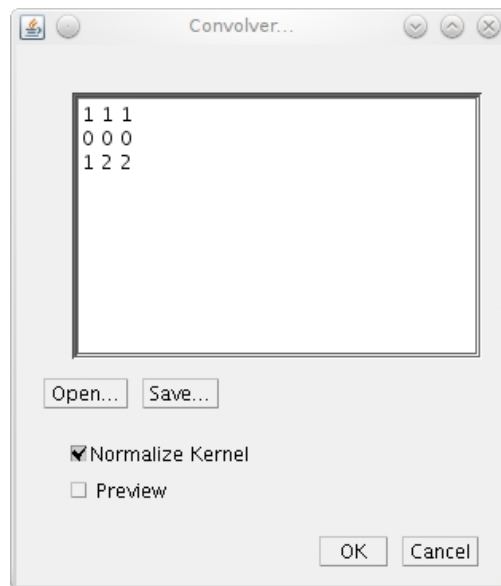
Filtrem liniowym nazywamy funkcję, która spełnia warunki liniowości tj. dla obrazów A i B oraz przekształcenia opisującego filtr α , spełnione są następujące własności:

1. Addytywność: $\alpha(A + B) = \alpha(A) + \alpha(B)$
2. Jednorodność: $\alpha(\lambda A) = \lambda\alpha(A)$, gdzie λ jest pewnym skalarzem.

3.1 Filtry uśredniające

Filtry uśredniające są realizowane przez maski, które pozwalają na uzyskanie średniej arytmetycznej lub ważonej jasności analizowanego punktu i jego otoczenia. Efektem działania filtrów uśredniających jest najczęściej usunięcie szczegółów obrazu i pozostawienie ogólnych kształtów obiektów. Takie filtry nazywamy filtrami dolnoprzepustowymi. Właściwe dopasowanie rzędu macierzy filtru pozwala na usunięcie z obrazu zakłóceń różnej wielkości.

Aby w *ImageJ* na obrazie zastosować filtry uśredniające należy wykorzystać do tego funkcję *Convolve* znajdującą się w menu *Process->Filters*. W oknie funkcji *Convolve* (Rysunek 5) możemy określić precyzyjnie maskę. Wymogiem jest, aby maska miała kształt kwadratowy oraz nieparzystą liczbę kolumn i wierszy. Zaznaczenie opcji *Normalize Kernel* wymusza podzielenie końcowej wartości piksela przez sumę wag macierzy filtru (normalizację), w przypadku gdy te wartości nie sumują się do 1.



Rysunek 5: Okno konstrukcji maski.

Do przeprowadzenia operacji uśredniających można zastosować następujące maski:

- Filtr rozmywający:
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Filtr dolnoprzepustowy: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
- Rozmycie gaussowskie: $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

3.2 Filtry wykrywające krawędzie

Zagadnienie wykrycia krawędzi obiektów na obrazie jest jednym z najważniejszych zagadnień analizy ilościowej. Jest bardzo wiele różnych algorytmów filtracji stosowanych do wykrywania krawędzi obiektów. Przez krawędź rozumiemy linię rozdzielającą obszary o różnym poziomie nasycenia czy jasności. Standardowo, krawędzie wykrywa się wykorzystując pochodne funkcji opisującej zmiany poziomu szarości obrazu wzdłuż prostej. Pochodne te, są jednak bardzo dobrze przybliżane przez gradienty, które mogą być interpretowane jako odjęcie od obrazu pierwotnego obrazu przesuniętego o jeden wiersz lub kolumnę. Gradienty można również wyznaczać jako różnice wartości sąsiednich punktów na obrazie. Do podstawowych filtrów gradientowych zaliczamy:

- Maskę Prewitta

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Maskę Sobela

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Maskę Robertsa

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

Maski są reprezentowane przez dwa rodzaje macierzy, ponieważ do wykrywania krawędzi w pionie i poziome stosuje się odrębne maski. Inne kierunki przeszukiwania krawędzi można uzyskać przez obrót wartości macierzy wokół jej punktu centralnego.

Znanym mechanizmem wykrywania krawędzi jest również tzw. operator Laplace'a:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Operator Laplace'a jest czuły na szum i może powodować wykrywanie fałszywych krawędzi. Żeby uniknąć tego efektu, często przez zastosowaniem maski Laplace'a stosuje się rozmycie gaussowskie obrazu. Dzięki własności operacji splotu maski Laplace'a i rozmycie gaussowskie można połączyć stosując do tego celu jedną macierz wag:

$$\begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

Do wykrywania krawędzi można również stosować filtry wyostrzające (tzw. górnoprzepustowe). Wadą tego typu filtrów jest fakt, iż powoduje on wyostrenie nie tylko krawędzi obiektów na obrazie, ale także szumów. Przykładem filtru wyostrzającego może być następująca maska:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

W *ImageJ* filtry wykrywające krawędzie można zastosować tak samo jak filtry uśredniające, czyli za pomocą okna `Convolve`.

4 Filtry nieliniowe

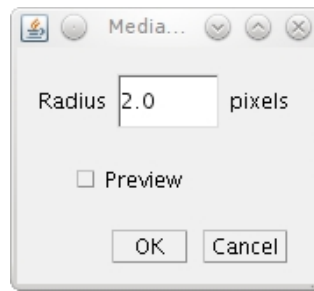
O części filtrów nieliniowych mówiliśmy wcześniej, przy okazji omawiania klasycznych funkcji nieliniowych zmieniających kontrast obrazu, tj. o funkcji logarytmicznej, wykładniczej, potęgowej.

Jedną z wad filtrów liniowych jest niszczenie drobnych szczegółów i krawędzi obrazów podczas usuwania szumów. Nieliniowe filtry kontekstowe, oparte na sąsiedztwie analizowanego punktu, posiadają znacznie większe możliwości niż filtry liniowe np. dokładniej lokalizują krawędzie niż laplasjany. Filtry nieliniowe możemy podzielić na: medianowe i adaptacyjne.

4.1 Filtry medianowe

Podstawą działania filtrów medianowych jest wybieranie jednej z sąsiednich wartości przetwarzanego punktu, jako wartości wynikowej dla obrazu wyjściowego. Wybór tej wartości jest dokonywany wg. pewnej reguły, którą w przypadku filtru medianowego jest wyznaczenie mediany wartości punktów sąsiednich analizowanego piksela. Mediana to wartość środkowa dla zbioru wartości. W naszym przypadku zbiorem wartości jest wartość piksela i jego najbliższego otoczenia. Spośród wartości tego zbioru (załóżmy, że mamy ich 9), wybieramy taką która jest większa od czterech najmniejszych i równocześnie mniejsza od czterech największych.

Ponieważ filtracja medianowa, nie wprowadza do obrazu wynikowego żadnych wartości pikseli, które nie istniałyby wcześniej na obrazie wejściowym, nie jest wymagana żadna normalizacja wyników filtracji. Dodatkową zaletą filtrów medianowych, jest fakt, że nie powodują one pogorszenia krawędzi obiektów na filtrowanym obrazie. Filtry medianowe są określane jako *silne* ponieważ wartości znacząco odbiegające od pozostałych oraz szumy lokalne nie mają wpływu na ostateczny wynik filtracji.



Rysunek 6: Okno wyboru promienia filtru medianowego.

Aby przeprowadzić filtrację medianową w programie *ImageJ* należy wybrać opcję *Median* z menu *Process->Filters*. W oknie polecenia (Rysunek 6) mamy dodatkową możliwość określenia promienia sąsiedztwa punktu, dla którego szukamy mediany.

Warto zauważyć, że w menu *Process->Filters* są dostępne dodatkowe filtry, których zasada działania jest analogiczna jak w przypadku filtru medianowego. Różnicą jest regułą wybierania wartości, do obrazu wyjściowego, którą może być np. średnia (*Mean*), minimum (*Minimum*), maksimum (*Maximum*) czy wariancja (*Variance*) punktów sąsiedztwa.

Przykładem filtru medianowego może być również funkcja *Despeckle* z menu *Process->Noise*. Jest to tak zwane usuwanie kurzu, które polega na usunięciu punktów w sposób istotny różniących się od swoich sąsiadów. Jeżeli różnica średniej wartości pikseli jest znacząco różna od analizowanego piksela, to jest on zastępowany przez tą średnią, w przeciwnym wypadku jego wartość pozostaje bez zmian.

4.2 Filtry adaptacyjne

Podstawową różnicą pomiędzy filtrami medianowymi, a adaptacyjnymi jest fakt, iż poprzednie nie potrafiły wzmocnić krawędzi, jednocześnie rozmywają obszary poza krawędziami. Filtry adaptacyjne działają dwuetapowo:

1. Najpierw na podstawie pewnego parametru punkty na obrazie są klasyfikowane jako krawędzie, bądź punkty do nich nie należące. Przykładem parametru decydującego o takiej kwalifikacji punktu, może być np. wariancja stopni szarości otoczenia punktu.
2. Następnie, filtracja uśredniająca jest wykonywana dla tych obszarów, które nie zostały rozpoznane w pierwszym kroku jako krawędzie.

Przykładami filtrów adaptacyjnych jest filtr *Wienera* lub *Kalmana*. Nie posiadają one jednak w *ImageJ* swojej bezpośredniej implementacji, ponieważ mogą być stosowane do ograniczonej liczby obrazów i zagadnień i wymagają dużej wiedzy z zakresu własności różnorodnych przekształceń.

5 Program ćwiczenia

Ćwiczenie składa się z trzech części:

Część 1 (max dst):

1. Wczytaj obraz *Fluorecent Cells* i zaszum jego drugi kanał (*C2 Fluorecent Cells*) za pomocą opcji `Add specific noise..` z parametrem 15.
2. Zduplikuj zaszumiony obraz.
3. Wygładź kopię za pomocą filtru medianowego.
4. Odejmij od zaszumionego obrazu jego wygładzoną kopię.
5. Wynik poprzedniej operacji odejmij od zaszumionego obrazu.
6. Wyjaśnij co się stało z pierwotnie zaszumionym obrazem?

Część 2 (max db):

1. Wykorzystaj wybrane operacje filtrowania w celu jak najlepszego usunięcia szumów z obrazu z punktu 1 poprzedniego ćwiczenia.
2. Jakie widzisz różnice w działaniu filtrów? Dlaczego? Podaj przykłady co najmniej trzech porównań.

Część 3 (max bdb):

1. Postaraj się wykryć krawędzie w trzecim kanale obrazu *Fluorecent Cells* (*C3 Fluorecent Cells*). Do tego celu wykorzystaj maski zaproponowane w części instrukcji dotyczącej wykrywania krawędzi. Pamiętaj, że lepsze efekty możesz uzyskać nakładając na siebie (a może mieszając) wyniki dla dwóch rodzajów filtracji (pionowej i poziomej). Możesz też zastosować inne maski, znalezione w sieci.

Hanna Kamińska