

Odszumianie obrazów

Seminarium: Algorytmy numeryczne i graficzne

Karolina Jeziorska, Oskar Bujacz

Wrocław, 8 czerwca 2020 r.

Spis treści

| | | |
|----------|---|-----------|
| 1 | Wstęp | 2 |
| 2 | Rodzaje szumów | 2 |
| 3 | Ocena jakości | 3 |
| 4 | Filtry ogólnie | 4 |
| 4.1 | Definicja | 4 |
| 4.2 | Splot | 5 |
| 5 | Filtry w dziedzinie przestrzennej | 6 |
| 5.1 | Filtry lokalne liniowe | 6 |
| 5.1.1 | Średnia | 6 |
| 5.1.2 | Gauss | 7 |
| 5.2 | Filtry lokalne nieliniowe | 7 |
| 5.2.1 | Mediana | 8 |
| 5.2.2 | Konserwatywne wygładzanie | 8 |
| 5.2.3 | Strojony filtr medianowy | 9 |
| 5.3 | Filtry nielokalne | 9 |
| 5.3.1 | Nielokalna średnia | 9 |
| 5.4 | Problem krawędzi | 9 |
| 6 | Filtry w dziedzinie częstotliwości | 10 |
| 6.1 | Transformacja Fouriera | 10 |
| 6.2 | Dyskretna transformacja Fouriera dwuwymiarowa | 11 |
| 6.3 | Filtrowanie w dziedzinie częstotliwości | 11 |
| 6.4 | Kroki filtrowania | 13 |
| 6.5 | Przykłady filtrów | 13 |
| 6.5.1 | Idealny filtr dolnoprzepustowy | 13 |
| 6.5.2 | Filtr Butterworth’a dolnoprzepustowy | 14 |
| 6.5.3 | Gaussowski filtr dolnoprzepustowy | 14 |
| 7 | Aktualne trendy | 16 |
| 8 | Zastosowanie | 16 |

1. Wstęp

W prawdziwym świecie sygnały nie są idealne, co więcej, zawierają one jakieś konkretne odstępstwa od takiego idealnego sygnału. Takie różnice nazywamy szumem. Szum powstaje podczas produkcji, tworzeniu, czy utrwalaniu prawdziwego sygnału. Nie jest jego częścią i może być spowodowany wieloma czynnikami: różnice w czułości urządzenia przechwytyjącego sygnał, zmiany otoczenia, błędy transmisji itd. Możemy także chcieć traktować nieistotne elementy obrazu jako szum (np. fakturę). Jednak wszystko sprowadza się do jednego — chcemy usunąć szum zachowując przy tym, jak najbardziej oryginalny sygnał, bez pogorszenia jego jakości.

Mówiąc o obrazach, myślimy raczej o obrazach kolorowych, czyli zamiast jednej macierzy opisującej obraz mamy minimum trzy. Pokazane w tej notatce filtry będą odnosić się do obrazów w skali szarości — mamy więc jedną macierz, która opisuje jasności pikseli, które przyjmują wartości z zakresu $[0, 255]$. Filtry w kolorze wykonuje się analogicznie do tych tutaj pokazanych, jednak wymagają one zazwyczaj przejście na inną przestrzeń barw (YUV). Spowodowane jest to tym, że ludzkie oko jest różnie czułe na różne barwy.

2. Rodzaje szumów

Szum możemy podzielić na dwa ogólne rodzaje:

- **szum niezależny** może być określony za pomocą modelu addytywnego, gdzie $f(x, y)$ to otrzymany obraz, $s(x, y)$ obraz prawdziwy a $n(x, y)$ to szum:

$$f(x, y) = s(x, y) + n(x, y) \quad (1)$$

- **szum zależny od danych na obrazie** może być wyrażone za pomocą bardziej skomplikowanych modeli (multiplikatywnych lub nieliniowych), dlatego raczej zakładamy, że szum jest niezależny

Przykładowe konkretne typy szumów to:

- **Sól i pieprz** nazywany inaczej też szumem impulsywnym, powodowany jest raczej przez błędy w przesyłaniu danych. Zepsute piksele są albo całkowicie białe lub czarne, nadając wygląd rozsypanej soli i pieprzu. Reszta pikseli zostaje niezmieniona. Często określany za pomocą procentu ile pikseli zostało zepsutych.



- **Szum Gaussa** to typowy szum, który powstaje podczas robienia zdjęcia. Wartości funkcji szumu mają rozkład Gaussa, czyli ich funkcja gęstości zadana jest wzorem:

$$f_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$



- **Szum okresowy** jest powodowany przez zakłócenie elektryczne lub elektromagnetyczne zachodzące podczas robienia zdjęcia. Wygląda, jakby na obraz nałożono powtarzający się wzór.



źródło: <https://hal.inria.fr/hal-01206546/file/ScipyLectures-simple.pdf>

3. Ocena jakości

Istnieje wiele narzędzi, by ocenić wpływ szumu na obraz. Jednak większość (jak nie wszystkie) potrzebują obrazu oryginalnego (bez szumów) lub wartości samego szumu. Można to rozwiązać poprzez szukanie obszarów niezaszumionych lub nakładanie szumu i późniejsze sprawdzanie jakości. Jednak dla ludzi ważnym sensem jest wzrok, który określa czy obraz dla nas jest zaszumiony, czy też nie — nie potrzebujemy do tego oryginalnego obrazu.

Wybrane metody oceny jakości obrazu (dla obrazu o wymiarach $M \times N$):

- **RMSE - root mean square error**, czyli pierwiastek błędu średniokwadratowego:

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=x}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - t(x, y)]^2} \quad (3)$$

- **SNR - signal to noise ratio**, czyli stosunek sygnału do szumu:

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - t(x, y)]^2} \quad (4)$$

- **PSNR - peak signal to noise ratio**, czyli szczytowy stosunek sygnału do szumu:

$$PSNR = \frac{[\max_{x,y} f(x, y)]^2}{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - t(x, y)]^2} \quad (5)$$

SNR i PSNR często podawane są w decybelach, które są logarytmiczną skalą porównawczą, czyli dokładnie mówią, o ile coś jest większe, mocniejsze od jakiegoś stanu zerowego.

Wartość wyrażona w decybelach to iloraz danej wartości R do pewnej wartości odniesienia R_0 :

$$R_{dB} = 10 \log_{10} \frac{R}{R_0} \quad (6)$$

4. Filtry ogólnie

4.1. Definicja

Filtry cyfrowe to ogólnie mówiąc algorytmy, które przekształcają jeden obraz w inny zgodnie z funkcją przejścia, która może być liniowa lub nie. Przekształcenie te jest dokonywane *piksel po pikselu*. W praktyce filtry wykorzystuje się w celach:

- stłumienie niepożądanego szumu
- poprawy jasności
- usunięcia określonych wad
- poprawy obrazu o złej jakości technicznej
- rekonstrukcji obrazu, który uległ częściowemu zniszczeniu

Patrząc na filtry jako na funkcje częstotliwości, można je podzielić na podstawie wartości, jakie przepuszczają:

- **Filtr dolnoprzepustowy (low pass)** tłumi składowe o dużej częstotliwości, a pozostawia bez zmiany te o małej. Ogólnie wygładzają obraz, uśredniają gwałtowne zmiany w intensywności. Służą głównie do wygładzenia obrazu i usunięcia szumu i zakłóceń, jednak przesadzenie może prowadzić zatracenia szczegółów i zbytniego rozmycia.
- **Filtr górnoprzepustowy (high pass)** jest używany do wzmocnienia szczegółów o dużej częstotliwości występujących w obrazie. Zwiększa się ostrość obrazu, przy czym wzmacnia się też szum. Części obrazu o większych częstotliwościach zostaną wzmocnione, a partie o małych zostaną wytłumione. Takie filtry służą do podkreślenia elementów, które charakteryzują się szybkimi i ostrymi zmianami jasności — kontury, krawędzie, drobne elementy, faktury. Dlatego obszary, w których występują duże zmiany jasności pikseli, są intensyfikowane, inne natomiast tonowane, tłumione. Na obrazie wynikowym prawie wszystkie punkty tła i leżące wewnątrz jakichś obiektów zostaną usunięte (wyzerowane), natomiast punkty należące do krawędzi i granic pomiędzy obiektami i tłem zostaną odpowiednio wyeksponowane i wyostrzone. Dlatego, mimo że mogą poprawić obraz poprzez wyostrenie szczegółów, przesadzenie może znacznie obniżyć jakość.

4.2. Splot

Splot (konwolucja) to działanie określone dla dwóch funkcji, które daje w wyniku inną. Mówi nam, jak kształt jednej funkcji jest modyfikowany przez drugą. Działanie to określa się wzorami:

$$h(t) = (f * g)(t) = \int_{-\infty}^{+\infty} f(t - \tau)g(\tau)d\tau \quad (7)$$

$$h[n] = (f * g)[n] = \sum_{m=-\infty}^{+\infty} f[m]g[n - m] \quad (8)$$

Dla funkcji dwóch zmiennych w dziedzinie dyskretnej splot wyraża się wzorem:

$$h[x, y] = (f * g)[x, y] = \sum_{n, m} w[n, m]j[x - n, y - m] \quad (9)$$

Funkcja ta ma następujące własności (wszystkie wynikają wprost z definicji):

- *przemienność* $f * g = g * f$
- *łączność* $(f * g) * h = f * (g * h)$
- *rozdzielność względem dodawania* $f * (g + h) = (f * g) + (f * h)$
- *łączność względem mnożenia przez skalary* $c(f * g) = (cf) * g = f * (cg)$

Twierdzenie o konwolucji dla transformat Fouriera Niech f i g będą dwoma funkcjami ze splotem $f * g$, a $\mathcal{F}\{f\}$ i $\mathcal{F}\{g\}$ to transformaty Fouriera tych funkcji. Wtedy zachodzi:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\} \quad (10)$$

Dowód:

Weźmy dwie funkcje całkowalne f, g . F i G zadane są wzorami:

$$F(v) = \mathcal{F}\{f\}(v) = \int_{\mathbb{R}^\times} f(x)e^{-2\pi i x \cdot v} dx \quad (11)$$

$$G(v) = \mathcal{F}\{g\}(v) = \int_{\mathbb{R}^\times} g(x)e^{-2\pi i x \cdot v} dx \quad (12)$$

Niech h będzie konwolucją f i g .

$$h(z) = \int_{\mathbb{R}^\times} f(x)g(z - x)dx \quad (13)$$

Transformacja Fouriera H jest zdefiniowana jako:

$$H(v) = \mathcal{F}\{h\}(v) = \int_{\mathbb{R}^\times} h(z)e^{-2\pi i z \cdot v} dz = \int_{\mathbb{R}^\times} \int_{\mathbb{R}^\times} f(x)g(z - x)e^{-2\pi i z \cdot v} dz = \quad (14)$$

$$\int_{\mathbb{R}^\times} f(x) \left(\int_{\mathbb{R}^\times} g(z - x)e^{-2\pi i z \cdot v} dz \right) dx \quad (15)$$

Stosując podstawienie $y = z - x$ i $dy = dz$ mamy:

$$H(v) = \int_{\mathbb{R}^\times} f(x) \left(\int_{\mathbb{R}^\times} g(y)e^{-2\pi i(y+x) \cdot v} dy \right) dx = \int_{\mathbb{R}^\times} f(x)e^{-2\pi i x \cdot v} \left(\int_{\mathbb{R}^\times} g(y)e^{-2\pi i y \cdot v} dy \right) dx = \quad (16)$$

$$\int_{\mathbb{R}^\times} f(x)e^{-2\pi i x \cdot v} dx \int_{\mathbb{R}^\times} g(y)e^{-2\pi i y \cdot v} dy \quad (17)$$

Czyli mamy:

$$H(v) = F(v) \cdot G(v) \quad (18)$$

Co kończy dowód.

Po nałożeniu transformaty odwrotnej \mathcal{F}^{-1} dostajemy:

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\} \quad (19)$$

Analogicznie można pokazać wszystko dla przypadku dyskretnego.

5. Filtry w dziedzinie przestrzennej

Mechanika filtrowania w dziedzinie przestrzennej polega na przesuwaniu maski filtra z punktu do punktu na obrazie. Dla każdego punktu (x, y) na obrazie wartość tego punktu zależy od wybranej funkcji przejścia. Rozróżniamy filtry lokalne i nielocalne oraz liniowe i nieliniowe.

W ogólności, filtrowanie liniowe obrazu f maską filtrującą w o wymiarze $m \times n$ wyraża się wzorem:

$$g(x, y) = \sum_{p=-a}^a \sum_{q=-b}^b f(x+p, y+q)w(p, q) \quad (20)$$

Gdzie $a = (m-1)/2$, a $b = (n-1)/2$. Ogólnie obszar maski będziemy nazywać dla uproszczenia oknem K . Wzór odpowiada wzorowi na spłot funkcji dwóch zmiennych w dziedzinie dyskretny dlatego filtrowanie liniowe w dziedzinie przestrzennej można nazywać *konwolucją maski z obrazem*.

Można uprościć zapis i powiedzieć, że dla każdego punktu R obrazu jego wartość to:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i \quad (21)$$

Gdzie w to współczynniki maski, a z to wartości odpowiadających im pikseli na wybranym obszarze.

Należy też pamiętać, że współczynniki powinny sumować się do 1 by uniknąć rozjaśnienia lub ściemnienia obrazu. Dlatego podany wyżej wzór zapisuje się częściej:

$$g(x, y) = \frac{\sum_{p=-a}^a \sum_{q=-b}^b f(x+p, y+q)w(p, q)}{\sum_{p=-a}^a \sum_{q=-b}^b w(p, q)} \quad (22)$$

5.1. Filtry lokalne liniowe

5.1.1. Średnia

Najprostszym filtrem liniowym jest filtr uśredniający. Idea za nim jest prosta: poprzez zamianę wartości każdego piksela na średnią z pikseli sąsiadujących dostajemy obraz ze zredukowanymi ostrymi przejściami między kolorem pikseli. Jako że szum to właśnie takie ostre przejścia to zwykłym uśrednieniem możemy go dobrze zredukować. Jednak brzegi też charakteryzują się dużą różnicą między pikselami, dlatego uzyskamy też niechciany efekt rozmycia krawędzi. Wielokrotne zastosowanie takiego filtra potrafi całkowicie zatrzeć szczegóły.

| | | |
|-----|-----|-----|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

Przykładowa maska uśredniająca rozmiaru 3×3

Wynik działania przedstawionej maski można zapisać w poznanej wcześniej konwencji:

$$R = \sum_{i=1}^9 \frac{1}{9} z_i = \frac{1}{9} \sum_{i=1}^9 z_i$$

Czyli pokazaną maskę można zastąpić taką złożoną z samych jedynek i na końcu pomnożyć wynik przez współczynnik skalujący równy $\frac{1}{mn}$. Ogólnie filtry uśredniające mogą używać średniej ważonej, czyli różne piksele w sąsiedztwie mogą być różnie ważne. Jednak wtedy należy pamiętać o skalowaniu wyniku.

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

Zmieniona przykładowa maska uśredniająca rozmiaru 3×3

Rozmycie przy zastosowaniu filtra uśredniającego jest proporcjonalne do wielkości maski — określa, jakiego rozmiaru obiekty zostaną *utopione* w tło.

W celu zmniejszenia negatywnych skutków zwykłego filtra uśredniającego stosuje się takie maski, gdzie środek (czyli aktualny piksel) ma większą wagę. Gdy w poniższej masce wartość $a > 1$ to wpływ pikseli sąsiadujących jest odpowiednio mniejszy.

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | a | 1 |
| 1 | 1 | 1 |

Przykładowa maska uśredniająca rozmiaru 3×3 z inną wagowością środkowego piksela

5.1.2. Gauss

Inną odmianą średniej ważonej jest zastosowanie współczynników rozkładu Gaussa. Wtedy wartości w_i to po prostu wartości funkcji:

$$w(p, q) = \frac{1}{2\pi\sigma^2} e^{-\frac{p^2+q^2}{2\sigma^2}} \quad (23)$$

Czyli zastosowanie filtra Gaussa to konwolucja obrazu z dwuwymiarową funkcją Gaussa.

Teoretycznie, wartości maski będą niezerowe dla całego obrazu, więc maska powinna być rozmiaru całego obrazu. Jednak w praktyce piksele w odległości większej niż 3σ od piksela środkowego mają na tyle mały wpływ by rozważać je jako zerowe. Widzimy więc, że rozmiar maski zależy od wartości σ i powinien wynosić $6\sigma \times 6\sigma$. Filtr Gaussa jest rozdzielny - zamiast stosować go raz dwuwymiarowo, można zastosować dwa razy wersję jednowymiarową: najpierw horyzontalnie a później wertykalnie.

| | | |
|------------|------------|------------|
| 0.00719253 | 0.05314604 | 0.00719253 |
| 0.05314604 | 0.3926991 | 0.05314604 |
| 0.00719253 | 0.05314604 | 0.00719253 |

Przykładowy filtr Gaussa, $\sigma = 0.5$

5.2. Filtry lokalne nieliniowe

Filtry nieliniowe również operują na sąsiedztwie punktu i przesuwaniu maski. Jednak wynik nie jest liniową kombinacją wartości jak w filtrach liniowych, a zależy w jakiś warunkowy sposób od wartości pikseli. Wynik

zależy od ustawienia (rankingu) pikseli sąsiadujących. Filtr zamienia wartość piksela środkowego z wartością piksela wybranego poprzez ustalony ranking. W ogólności filtrowanie lokalne nieliniowe wyraża się wzorem:

$$g(x, y) = \phi(f(x, y)) \quad (24)$$

Gdzie zbiorem wartości funkcji ϕ są wartości pikseli z otoczenia punktu (x, y) .

5.2.1. Mediana

Najbardziej znanym filtrem w tej kategorii jest filtr medianowy, który zastępuje wartość piksela poprzez medianę pikseli sąsiadujących (oczywiście oryginalna wartość piksela też w liczeniu mediany jest uznawana). Filtry medianowe bardzo dobrze działają w przypadku szumu impulsywnego, czyli soli i pieprzu. Wzór na wartość piksela (x, y) można więc zapisać jako:

$$g(x, y) = \text{median}_{p,q \in K}(f(x + p, y + q)) \quad (25)$$

Gdzie K było już wcześniej określone jako obszar filtra.

Głównym celem filtrów medianowych jest zmuszenie punktów o wyróżniających się poziomach szarości do bycia bardziej jak sąsiedzi. Prowadzi to jednak do powstawania obszarów o stałej jasności bez jakiegokolwiek faktury. Jeżeli mamy wyizolowane zbiory pikseli, które są stosunkowo jasne lub ciemne o powierzchni mniejszej niż połowa filtra to zostaną one wyeliminowane. Filtr medianowy nie powoduje takiego rozmycia jak wcześniej przedstawione filtry liniowe. Filtracja medianowa nie wprowadza dodatkowych wartości do obrazu, dlatego nie wymaga skalowania.

Jednym z problemów tej metody jest długi czas obliczeń — należy posortować punkty, by znaleźć ich środkowy element. Sortowanie wykonuje się dla każdego użycia filtra, więc dla każdego piksela na obrazie. Przy małej masce nie jest to problemem, ale warto zauważyć, że czas wykonania takiego filtra rośnie wraz z rozmiarem maski. Dlatego czasami stosuje się różne zabiegi, by tego sortowania uniknąć. Można wykorzystać posortowanie pikseli z poprzedniego punktu dla sortowania dla aktualnego albo zastosować okno w innym kształcie i rozmiarze (często stosuje się okno w kształcie krzyża). Można też stosować specjalne wzory do obliczenia mediany:

$$MED(a, b, c, d, e) = \max(\min(a, b, c), \min(a, b, d), \min(a, b, e), \min(a, c, d), \min(a, c, e), \quad (26)$$

$$\min(a, d, e), \min(b, c, d), \min(b, c, e), \min(b, d, e), \min(c, d, e)) \quad (27)$$

Lub zastosować uproszczoną wersję filtra - filtr pseudomedianowy:

$$MED(a, b, c, d, e) = \frac{1}{2} \max(\min(a, b, c), \min(b, c, d), \min(c, d, e)) + \frac{1}{2} \min(\max(a, b, c), \max(b, c, d), \max(c, d, e)) \quad (28)$$

5.2.2. Konserwatywne wygładzanie

Inną możliwością jest użycie funkcji minimum i maksimum. Wyliczenie tych wartości jest dużo szybsze niż sortowanie i wybieranie mediany. Taki filtr opiera się na tym, że wartość piksela nie musi być uśredniona, ale musi znajdować się w pewnym zakresie wyznaczonym przez jego otoczenie. Dlatego wyliczamy $\min = \min_{p,1 \in K} f(x + p, y + 1)$ i $\max_{p,q \in K} f(x + p, y + q)$ z pominięciem $p = q = 0$. Wtedy:

$$g(x, y) = \begin{cases} \min & f(x, y) < \min \\ \max & f(x, y) > \max \\ f(x, y) & \text{wpp} \end{cases} \quad (29)$$

Taki filtr ponownie dobrze usuwa szum impulsywny i dobrze zachowuje krawędzie, jednak efekt jego działania jest dużo subtelniejszy.

5.2.3. Strojony filtr medianowy

Jest to wariant filtru medianowego, a tak naprawdę pewnego rodzaju połączenie filtru medianowego i uśredniającego. Z otrzymanego ciągu będziemy usuwać $d/2$ wartości najmniejszych i $d/2$ wartości największych z okna K o rozmiarach $n \times n$. Filtr, który otrzymamy po uśrednieniu wyraża się wzorem:

$$g(x, y) = \frac{1}{n^2 - d} \sum_{p, q \in K} f(x + p, y + q) \quad (30)$$

Gdzie $d = 0, \dots, n^2 - 1$.

Zauważmy, że dla $d = 0$ dostajemy zwykły filtr uśredniający, a dla $d = n^2 - 1$ filtr medianowy. Dla pozostałych wartości d otrzymujemy wartości pośrednie, więc filtr może być przydatny, gdy mamy różne typy szumów.

5.3. Filtry nielokalne

Tak jak mamy filtry lokalne, które do wyznaczenia wartości piksela wykorzystywały piksele z jego najbliższego otoczenia, tak mamy też filtry nielokalne, które wykorzystują wartości pikseli całego obrazka.

5.3.1. Nielokalna średnia

W przeciwieństwie do średniej lokalnej, żeby określić wartość danego piksela, bierzemy średnią ze wszystkich pikseli na obrazku. Jednak wagi tych pikseli są zależne od tego, jak podobne są do punktu (x, y) . Czystość obrazu, jak i zachowanie krawędzi jest dużo lepsze niż w zwykłej średniej.

Algorytm nielokalnej średniej dla obrazu o wymiarach $N \times M$ wyraża się wzorem:

$$g(x, y) = \frac{1}{C(x, y)} \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} f(p, q) W(x, y, p, q) \quad (31)$$

Gdzie $C(x, y)$ to wartość skalująca:

$$C(x, y) = \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} W(x, y, p, q) \quad (32)$$

Funkcja wagowa W określa podobieństwo dwóch punktów na podstawie ich otoczenia. Stosuje się różne funkcje wagowe, przykładowo może to być funkcja wagowa Gaussa:

$$W(x, y, p, q) = e^{-\frac{(B(x, y) - B(p, q))^2}{h^2}} \quad (33)$$

Gdzie h to parametr filtru, a funkcja B zwraca lokalną średnią punktu (czyli średnią z jego otoczenia).

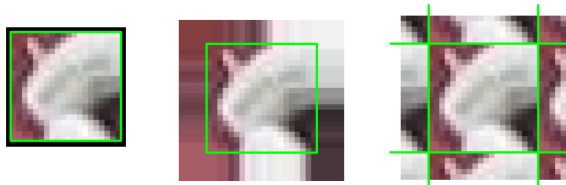
Liczenie wartości tego filtra może być bardzo kosztowne, dlatego pewnym sposobem jest ograniczenie pola maski.

5.4. Problem krawędzi

Oczywistym problemem do rozpatrzenia przy stosowaniu filtrów przestrzennych jest sytuacja, gdy centrum filtra zbliża się do krawędzi obrazu. Rozważmy kwadratową maskę rozmiaru $n \times n$. Gdy środek maski będzie w odległości mniejszej niż $(n-1)/2$ od krawędzi obrazu to część maski znajdzie się poza obrazem. Z podanym problemem można poradzić sobie na kilka sposobów:

- ograniczenie obszaru działania maski tylko na taki, by nigdy nie znalazła się poza obrazem. Czyli nakładamy ograniczenie tak, by środek maski był przynajmniej $(n-1)/2$ pikseli od krawędzi obrazu. Finalny obraz będzie mniejszy (jeśli go obetniemy) lub będzie miał nieprzetworzone krawędzie. Jeżeli brzegi są nieważne lub maska jest mała, to jest to najlepsze rozwiązanie.

- filtrowanie pikseli brzegowych tylko tą częścią maski, która zawiera się w obrazie.
- zwiększenie obszaru wyjściowego poprzez dodanie *paddingu*, czyli uzupełnieniu poza brzegi (wartościami zerowymi - kolor czarny, powtarzając brzegi czy też zawijając cyklicznie obraz). Jednak wtedy dodane uzupełnienie i jego rodzaj wpłynie na wartość pikseli brzegowych.



źródło: <http://aragorn.pb.bialystok.pl/> boldak/

6. Filtry w dziedzinie częstotliwości

Mimo wielu prób, metody odszumiania w dziedzinie przestrzennej wciąż posiadają pewne wady. Stosowanie masek prowadzi to nieuniknionej utraty szczegółów, pojawiają się problemy z wykrywaniem krawędzi, dlatego też podjęto próby zminimalizowania tego zjawiska przez np. ADF (Anisotropic Diffusion Filtering). Niestety wtedy pojawia się niepożądany efekt schodków. Dlatego narodził się pomysł zmiany dziedziny, w której operujemy - z przestrzennej na częstotliwości. Pojawiło się mnóstwo różnych transformacji, które się tym zajmują: Są to np.: transformacja Fouriera, jej szybka wersja (FFT), dyskretna transformacja cosinusowa, transformacje korzystające z falek, falek sterujących, curvelets, ridgelets, wedgelets i inne. Angielskie nazwy są tu nieprzypadkowe, gdyż nie mamy precyzyjnych odpowiedników w języku polskim. Skupimy się na transformacji Fouriera ze względu na całkiem niezłe efekty przy jej zastosowaniu oraz przynajmniej pobieżną znajomość słuchaczy z jej właściwościami.

6.1. Transformacja Fouriera

Z uwagi na poruszenie tego tematu na jednym z seminarium, przypomnimy tylko najważniejsze informacje. Jednowymiarową transformatę Fouriera $F(u)$ funkcji jednej zmiennej $f(x)$ określamy jako

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi ux} dx \quad (34)$$

gdzie $i^2 = -1$ natomiast $e^{-i2\pi ux}$ to postać wykładnicza pewnej liczby zespolonej. Wiemy również, że możemy otrzymać $f(x)$ stosując odwrotną transformację Fouriera.

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{i2\pi ux} du \quad (35)$$

Te dwa równania tworzą razem parę transformacji Fouriera. Oznacza to, że po zastosowaniu jednej z nich na pewnym wektorze v i otrzymaniu v' , możemy użyć drugiej, aby wrócić do v . Dowód ten był pokazany przy okazji interpolacji trygonometrycznej, zatem nie będziemy go powtarzać.

Łatwo możemy zastosować tą transformację do przypadku dwuwymiarowego. Jest ona wtedy określona jako

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-i2\pi(ux+vy)} dx dy \quad (36)$$

a jej odwrotność to

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(ux+vy)} du dv \quad (37)$$

Powyższe rozważania dotyczą przypadku ciągłego, ale w przypadku odszumiania, gdzie mamy dyskretne wartości poszczególnych pikseli, możemy skupić się na transformacji dyskretnej.

Dyskretna transformacja Fouriera będzie podstawą dalszych rozważań. Dla funkcji $f(x)$ oraz długości wektora M mamy

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-i2\pi ux/M} \quad \text{dla } u = 1, \dots, M-1 \quad (38)$$

Podobnie jak w przypadku ciągłym, podamy odwrotność

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{i2\pi ux/M} \quad \text{dla } x = 1, \dots, M-1 \quad (39)$$

Ważną własnością dla przykładu dyskretnego jest fakt, że zawsze będzie istnieć odwrotność transformacji w naszym przypadku, gdy mamy skończone wartości funkcji.

Kolejnym spostrzeżeniem jest miejsce czynnika $\frac{1}{M}$, który możemy umieścić dowolnie. Istotne jest jedynie to, aby iloczyn współczynników znajdujących się przed sumą w $F(u)$ oraz $f(x)$ był równy właśnie $\frac{1}{M}$, czyli odwrotność długości wektora.

Korzystając z wzoru Eulera $e^{i\theta} = \cos \theta + i \sin \theta$ oraz parzystości cosinusa rozpiszmy wzór na $F(u)$

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) (\cos(2\pi ux/M) - j \sin(2\pi ux/M)) \quad \text{dla } u = 1, \dots, M-1 \quad (40)$$

Widzimy, że każdy wyraz w dziedzinie częstotliwości wyrażony jest przez liniową kombinację wszystkich innych wartości funkcji z dziedziny przestrzennej. Współczynniki \sin oraz \cos odpowiadają różnym częstotliwościom.

6.2. Dyskretna transformacja Fouriera dwuwymiarowa

Przejdźmy teraz do narzędzia, na którym będziemy się opierać. Dla funkcji dwóch zmiennych $f(x, y)$ oraz odpowiednio M i N wartości określamy DFT 2D następująco

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(ux/M+vy/N)} \quad (41)$$

Podobnie jak w 1D $u = 0, \dots, M-1$ oraz $v = 0, \dots, N-1$. Bardzo podobnie opisujemy transformację odwrotną

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(ux/M+vy/N)} \quad (42)$$

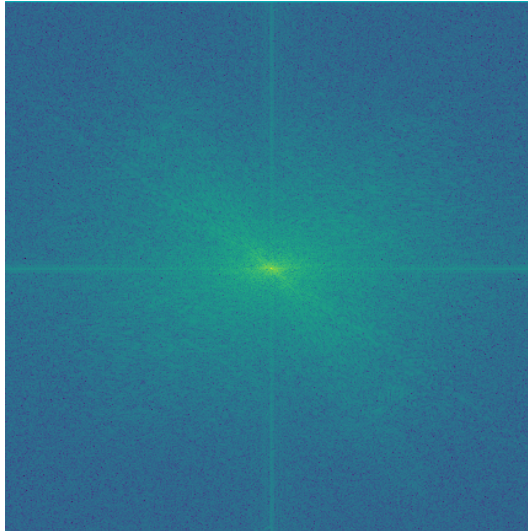
6.3. Filtrowanie w dziedzinie częstotliwości

W równaniu (40) widzimy, że każdy wyraz $F(u, v)$ zawiera wszystkie wartości $f(x, y)$ zmodyfikowane przez pewien czynnik wykładniczy. Wydawałoby się, że nie możemy zatem wyciągnąć wniosków o obrazie po zastosowaniu transformacji. Wiemy natomiast, że częstotliwość jest bezpośrednio powiązana z tempem zmian, zatem intuicyjnie możemy powiązać poszczególne częstotliwości ze wzorcami zmian w obrazie. Wprowadźmy przekształcenie S , które pozwala umieścić punkt odpowiadający ($u = v = 0$) na środku obrazu w punkcie $(u/2, v/2)$. Tą operację wykonujemy następująco



Rysunek 1: Lena w dziedzinie przestrzennej

- podział obrazu na 4 ćwiartki
- zamiana naprzeciwległych ćwiartek



Rysunek 2: Lena w dziedzinie częstotliwości

Ułatwia to intuicyjne zrozumienie rozkładu na składowe częstotliwości oraz nakładanie filtrów. Załóżmy, że zastosowaliśmy przesunięcie przekształcenie S celem ułatwienia interpretacji. W punkcie odpowiadającym $(u = v = 0)$, (teraz $(u/2, v/2)$) który odpowiada najwolniej zmieniającej się składowej częstotliwości, mamy wyraz określony wzorem

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{n-1} f(x,y) \quad (43)$$

Bez trudu interpretujemy wzór jako średnią intensywności z całego obrazu. W miarę oddalania się od środka obrazu po przekształceniu, zwiększają się poziomy zmian. Obszary blisko środka będą odpowiadać miejscom

stosunkowo gładkim. Weźmy przykład zdjęcia pokoju. Reprezentować będą to kawałki ścian czy jednolity kolor kanapy. Im dalej się oddalimy od środka, tym gwałtowniejsze zmiany w jasności wykryjemy, czyli wszelkie detale, krawędzie oraz szum.

6.4. Kroki filtrowania

W dziedzinie częstotliwości możemy wyróżnić następujące kroki filtrowania obrazu:

1. obliczamy $F(u, v)$ przez DFT
2. wykonujemy przekształcenie S w celu wyśrodkowania transformacji
3. mnożymy $F(u, v)$ przez filtr $H(u, v)$
4. ponownie wykonujemy S
5. obliczamy odwrotną DFT
6. zostawiamy tylko części rzeczywiste

Funkcję $H(u, v)$ nazywamy filtrem, ponieważ obniża on pewne częstotliwości, pozostawiając inne niezmienione

6.5. Przykłady filtrów

6.5.1. Idealny filtr dolnoprzepustowy

Najprostszym pomysłem na filtrowanie w dziedzinie częstotliwości jest wyzerowanie wszystkich składowych częstotliwości oddalonych o więcej niż ustalona odległość D_0 od punktu $(u/2, v/2)$. Zapiszemy go jako:

$$H(u, v) = \begin{cases} 1 & D(u, v) < D_0 \\ 0 & D(u, v) > D_0 \end{cases} \quad (44)$$

gdzie

$$D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2} \quad (45)$$

Powyższa funkcja zeruje częstotliwości w sposób symetryczny, ale można to również zrobić to niesymetrycznie, zamiast okręgu nałożyć elipsę. Uwzględnimy wtedy różnice w wymiarze pionowym i poziomym.

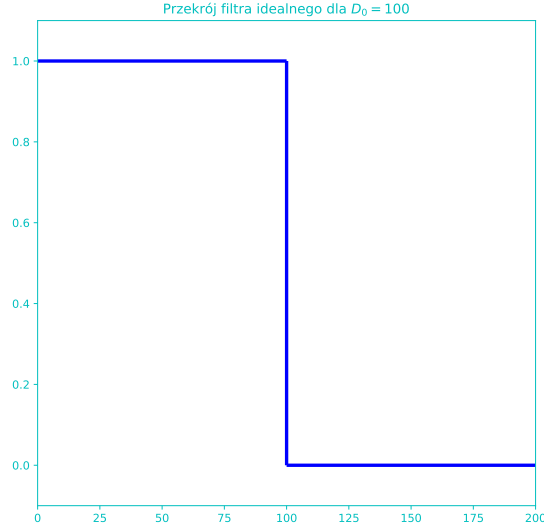
Problemem pozostaje oczywiście wybór odpowiedniego D_0 . Można to zrobić metodą prób i błędów, żeby później wizualnie, i przez jakiś wskaźnik, np. SNR ocenić jakość odszumiania. Z pomocą przychodzi tutaj widmowa gęstość mocy, która określa dystrybucję energii w poszczególnych składowych częstotliwościach naszego sygnału. Definiujemy ją jako

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u) \quad (46)$$

gdzie R i I to odpowiednio rzeczywista i urojona część liczby zespolonej
Zdefiniujemy energię całkowitą.

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v) \quad (47)$$

gdzie $P(u, v)$ mamy dane w równaniu 39. Jeśli transformacja została wyśrodkowana, czyli tak jak zakładamy w naszym przypadku, to możemy określić α jako



$$\alpha = 100 \sum_u \sum_v P(u, v) / P_T \quad (48)$$

która mówi jaki procent energii znajduje się w środku okręgu określonego przez D_0 . Sumujemy po tych punktach, które leżą w tym okręgu. Oczywiście, możemy stosować ten wzór do metod, które podamy poniżej. Filtr ten nie jest zbyt efektywny, ponieważ występuje zjawisko *ringingu*. Wizualnie można to opisać jako fale odbijające się od krawędzi na obrazie. W notebook'u znajdują się przykłady tego zjawiska. Na podstawie różnych wartości D_0 udało się wyciągnąć kilka wniosków. Komponenty centralne są odpowiedzialne za zbytne wygładzenie obrazu, natomiast kolejne koncentrycznie rozchodzące się częstotliwości odpowiadają za zjawisko ringingu. Przejdźmy do kolejnych metod, w których będziemy chcieli pozbyć się tych wad.

6.5.2. Filtr Butterworth'a dolnoprzepustowy

Filtr dolnoprzepustowy Butterworth'a rzędu n i promieniu D_0 definiujemy jako

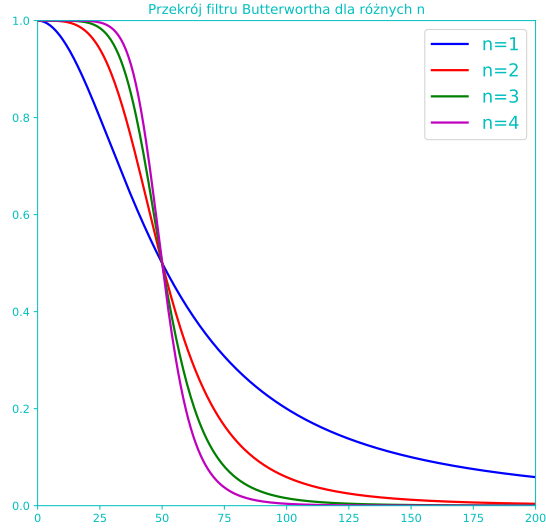
$$H(u, v) = \frac{1}{1 + (D(u, v)/D_0)^{2n}} \quad (49)$$

W porównaniu do poprzedniej metody filtrowania, częstotliwości nie są tu odcinane w jednym miejscu. Istnieje łagodne przejście między tymi, które są zachowywane w całości, a tymi które pomijamy. Odległość D_0 określa tutaj okrąg, na którym będziemy uwzględniali połowę mocy intensywności. Parametr n określa kształt krzywej w przekroju poprzecznym. Im wyższy parametr, tym bliżej do idealnego filtra dolnoprzepustowego. Uznaje się, że dla $n = 1$ i $n = 2$ nie widać zjawiska ringingu, jednak dla wyższych może być on już widoczny. W granicy dla $n \rightarrow \infty$ filtr ten jest tożsamy z idealnym filtrem dolnoprzepustowym.

6.5.3. Gaussowski filtr dolnoprzepustowy

Kolejnym filtrem jest filtr Gaussowski dolnoprzepustowy. Definiujemy go jako

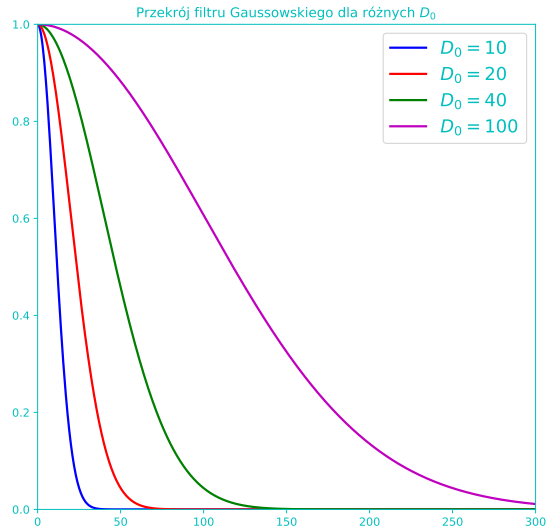
$$H(u, v) = e^{-D^2(u, v)/2\sigma^2} \quad (50)$$



σ jest parametrem, który mówi nam o kształcie krzywej w przekroju poprzecznym. Będziemy stosować $\sigma = D_0$, wtedy wzór wynosi

$$H(u, v) = e^{-D^2(u, v)/2D_0^2} \quad (51)$$

W przypadku tego filtra w odległości D_0 od środka będziemy zachowywać $\frac{2}{3}$ oryginalnej częstotliwości. Filtr ten powinien całkowicie eliminować zjawisko ringingu poprzez łagodne redukowanie kolejnych składowych.



7. Aktualne trendy

Od prostych filtrów lokalnych, poszukiwania podobieństw w innych obszarach w filtrach nielokalnych, próby wykorzystania gradientów do wykrywania krawędzi, przez transformację częstotliwości, aż do głębokich sieci neuronowych - bardzo trudno jest opisać wszystkie metody usuwające szum z obrazów. Kilka z używanych obecnie metod to:

- użycie rzadkiej i nadmiarowej reprezentacji na uczonych słownikach
- transformacja falkowa
- BM3D - block-matching and 3D filtering
- sieci neuronowe
- metody hybrydowe

Jedną z najbardziej efektywnych metod pod względem PSNR była metoda Multi-level Wavelet-Convolution Neural Network (MWCNN). Zrozumienie jej działania mocno wykracza poza temat dzisiejszego seminarium. Pokazuje to, jak długą drogę przeszło już zagadnienie odszumiania obrazów.

Dla zainteresowanych tematem odsyłam do repozytorium na GitHubie, które zbiera najlepsze metody wraz z implementacjami. Znajduje się tam również wspomniana MWCNN. Link poniżej.

<https://github.com/wenbihan/reproducible-image-denoising-state-of-the-art>

8. Zastosowanie

Odszumianie pojawia się w wielu aspektach dzisiejszego życia. Od tych ratujących życie, przez wykorzystania militarne i kosmiczne, do zdjęć w social-mediach. Kilka z nich zostało podanych poniżej.

- medyczne - zdjęcia rentgenowskie, tomografia komputerowa
- zdjęcia podwodne
- radary z syntetyczną aperturą, zdjęcia satelitarne
- zdjęcia w podczerwieni
- smartfony
- rozpoznawanie i klasyfikacja obrazów - preprocessing do ML, DL
- monitoring

Literatura

- [1] Witold Malina, Maciej Smiatacz *Cyfrowe przetwarzanie obrazów*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2008
- [2] Cezary Bołdak *Wykład: cyfrowe przetwarzanie obrazów* Politechnika Białostocka
- [3] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, B.S. Sohi, Apoorav Sharma *Image denoising review: From classical to state-of-the-art approaches* Information Fusion 55, 2020
- [4] Antoni Buades, Bartomeu Coll, Jean-Michel Morel *A review of image denoising algorithms, with a new one* Multiscale Modeling Simulation Volume 4 Issue 2, 2005

- [5] Rafael C. Gonzalez, Richard E. Woods *Digital Image Processing* Prentice Hall 2002
- [6] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>