

upel.agh.edu.pl

SW: Filtracja w dziedzinie częstotliwości

8 — 11 minut

Tomasz Kryjak, Piotr Pawlik

Filtracja w dziedzinie częstotliwości.

Cel:

- zapoznanie z filtracją w dziedzinie częstotliwości
- zapoznanie z metodami odzyskiwania obrazu z użyciem dekonwolucji

A. Filtracja obrazu w dziedzinie częstotliwości

Na jednym z poprzednich ćwiczeń zetknęliśmy się z pojęciem konwolucji - np. filtracja dolno i górnoprzepustowa. Operacja ta odpowiada mnożeniu w dziedzinie częstotliwości zgodnie z zależnością:

$$g(x,y)*f(x,y)=G(u,v)F(u,v)$$

(* oznacza konwolucję)

Kilka uwag:

- operacja filtracji w dziedzinie częstotliwości może okazać się bardziej efektywna w niektórych przypadkach - a dokładniej jeżeli operacje fft2 i ifft2 zajmą mniej czasu niż klasyczna konwolucja (zazwyczaj dla dużego obrazu, z dużą maską). Sama filtracja w dziedzinie częstotliwości to mnożenie całego obrazu przez jedną maskę
- w przypadku filtracji w dziedzinie częstotliwości zakłada się, że obraz "zawija się" na brzegach - co powoduje pewne artefakty

- filtracje w dziedzinie częstotliwości dotyczą tylko konwolucyjnych filtrów liniowych - filtry medianowe, maksymalne, minimalne itp. nie mają swoich odpowiedników.
1. Utwórz nowy m-plik. Na początku wykonaj polecenia `close all`; `clear all`;
 2. Wczytaj obraz "lena.bmp". Wykonaj transformatę Fouriera (łącznie z shiftem) dla obrazu "lena.bmp" - wykorzystaj stworzony poprzednio kod. Wyświetl obraz oryginalny, amplitudę i fazę.
 3. Przeprowadź filtracje dolnoprzepustową - usuń wysokie częstotliwości (czyli przy tak przyjętej wizualizacji leżące daleko od środka F-obrazu (amplitudy)).
 - na początku tworzymy filtr "kołowy", dolnoprzepustowy
 - polecenie `[f1,f2] = freqspace(512,'meshgrid')`; wygeneruje wektory opisujące przestrzeń w dziedzinie częstotliwości (512 - rozmiar obrazka)
 - `Hd = ones(512)`; - początkowo filtr inicjujemy samymi wartościami '1' 512 - rozmiar obrazka
 - `r = sqrt(f1.^2 + f2.^2)`; - wektor r promień koła w dziedzinie częstotliwości
 - `Hd((r>0.1)) = 0`; - usuwamy interesujący nas zakres częstotliwości zerując odpowiadające im miejsca filtru - w zależności co zostanie usunięte można uzyskać odpowiedni typ filtru (dolno, górno, pasmowoprzepustowy)
 - filtr można zwizualizować: `colormap(jet(64))`; `mesh(f1,f2,Hd)`;
 4. Wykonaj właściwą filtrację, czyli mnożenie F-obrazu przez filtr Hd (mnożenie macierzy typu element przez element).
 5. Wykonaj operację odwrotnego przesunięcia (`ifftshift`) i odwrotnej transformaty (`ifft2`), wynik wyświetl.
 6. Poeksperymentuj z rozmiarem filtru, zaimplementuj filtr górnoprzepustowy (odwrócenie warunku przy 'r') oraz pasmowo przepustowy (dwa warunki na 'r' połączone operatorem '|' (or) lub '&' (and)) - co najmniej takie trzy filtry należy zamieścić w m-pliku.

7. W ten sposób zaimplementowana filtracja wprowadza pewne artefakty - w postaci "pierścieni" wokół krawędzi. Zapobiec temu zjawisku można zapobiec odpowiednio "modelując" filtr - np. jako okno Hamming, Hanninga, Chebysheva (znane z przetwarzania sygnałów 1D).

8. Wykorzystaj instrukcje:

`h = fwind1(Hd,hanning(21));` - tworzenie filtra 2D z wykorzystaniem okna Hanninga

`[H f1 f2] = freqz2(h,512,512);` - wyznaczanie odpowiedzi filtra

Zamiast okna Hanninga można wykorzystać inne - funkcje : `hamming`, `chebwin` itp. więcej w dokumentacji do funkcji `fwind1`.

Okno można zwizualizować:

`mesh(f1,f2,H);`

9. Przeprowadź przykładową filtrację dolno i górnoprzepustową. Sprawdź w jaki sposób użycie okna poprawia rezultat filtracji. Pokaż rezultat prowadzącemu.

Metody odzyskiwania obrazu

B. Metoda filtra odwrotnego

Każdy system wizyjny podczas akwizycji sceny wnosi zniekształcenia i zakłócenia do obrazu. Problem ich eliminacji znany jest w literaturze pod pojęciem odzyskiwania obrazu (ang. *image restoration*). Matematyczny model obrazu zniekształconego przedstawia się następująco:

$$g(x,y)=f(x,y)*psf(x,y)+n(x,y)$$

C. Metoda filtra pseudoodwrotnego

Poprzednio szum został pominięty. Gdyby jednak należało go uwzględnić, to równanie opisujące obraz wynikowy przyjąłoby postać:

$$f(x,y)=|IDFT(G(u,v)PSF(u,v)-N(u,v)PSF(u,v))|f(x,y)=|IDFT(G(u,v)PSF(u,v)-N(u,v)PSF(u,v))|$$

Problemem jaki występuje w tym przypadku jest nieznajomość funkcji $N(u,v)$

1. Wczytaj obraz "inv_filter_noisy.mat" poleceniem load
inv_filter_noisy.mat oraz wyświetl go. Obraz jest uśredniony za pomocą maski współczynników *psf* stworzonej poprzez wywołanie fspecial('disk', 10); oraz dodatkowo zawiera szum. Poziom sygnału do szumu (ang. *Signal-to-noise ratio*) wynosi 70 dB. Założeniem w pierwszej fazie ćwiczenia jest brak szumu na obrazie (mimo, że on występuje).
2. Wykonaj operacje analogiczne do tych z punktu B (wyznacz F-obraz wczytanego obrazu oraz skonstruuj maskę współczynników *psf*, wyznacz F-obraz funkcji *psf*, podziel punktowo F-obraz wczytanego obrazu przez F-obraz maski *psf* oraz dokonaj wyznaczenia modułu z odwrotnej transformaty Fouriera tego wyrażenia). Czy obraz został odzyskany?
3. Wykonaj polecenie surf(fftshift(abs(PSF))), shading interp, które w rezultacie wyświetli odpowiedź impulsową filtra. Przyjrzyj się uważnie wykresowi oraz skalom na osiach. Wywnioskuj, dlaczego uzyskano taki efekt po odzyskaniu obrazu. Podpowiedź: przyjrzyj się wartościom maski *psf* oraz temu jak one wpływają część wzoru związaną z szumem.
4. Spróbujmy wyeliminować niekorzystne "działanie" bliskich zeru elementów maski *psf*. W tym celu w wyniku punktowego dzielenia F-obrazu przez PSF wyzerujemy wszystkie punkty, w których współczynniki maski *psf* są co do modułu mniejsze niż 0.01. Praktycznie można to zrealizować wykorzystując następującą składnię : $FN(abs(PSF) < 0.01) = 0$; gdzie FN -wyniki dzielenia F-obrazu przez PSF.
5. Następnie wyznacz moduł z odwrotnej transformaty Fouriera tego wyrażenia. Czy udało się odzyskać w pełni obraz?
6. Pokaż wyniki prowadzącemu.

D. Metoda filtracji Wienera

Metoda filtracji Wienera jest używana wtedy, kiedy znana jest

postać funkcji *psf* i poziom szumu albo można dokonać ich estymacji. Filtr Wienera poszukuje estymaty obrazu idealnego w oparciu o minimalizację funkcji błędu:

$$e^2 = E\{(f - \hat{f})^2\} \quad e^2 = E\{(f - \hat{f})^2\}$$

gdzie:

- $E\{\}$ - wartość oczekiwana wyrażenia,
- f - obraz odniesienia,
- \hat{f} - obraz estymowany

W dziedzinie częstotliwości estymata przyjmuje postać

$$\hat{F}(u,v) = [1 \text{PSF}(u,v) \cdot |\text{PSF}(u,v)|^2 / (|\text{PSF}(u,v)|^2 + 1 \text{SNR})] \cdot G(y,v)$$

gdzie: SNR - Signal-to-NoiseRatio - współczynnik mocy sygnału do mocy szumu

Odzyskany obraz jest modulem z odwrotnej transformaty Fouriera estymaty

$$f(x,y) = |\text{IDFT}(\hat{F}(u,v))|$$

Wartość SNR może być znana bezpośrednio jako iloraz mocy widma sygnału do mocy widma szumu (lub po prostu podana - jak w poniższym zadaniu). W niektórych przypadkach można też użyć wariancji - np. dla addytywnego szumu białego moc widma szumu można zastąpić przez jego wariancję. Wówczas SNR jest estymowana jako iloraz wariancji sygnału do wariancji szumu. Można też dla nieujemnego sygnału (a takim jest obraz) estymację SNR jako $12 \cdot \text{wariancja_sygnału}$. W ćwiczeniu trzeba będzie sprawdzić wszystkie trzy metody.

1. Wczytaj obraz "wiener_motion_noisy.png" poleceniem `imread` oraz wyświetl go. Uśrednienie obrazu dokonano za pomocą maski współczynników *psf* stworzonej poprzez wywołanie `fspecial('motion', 15, 0)`; Poziom sygnału do szumu (SNR) wynosi 70 dB. Podczas konwolucji założono ciągłość obrazu.
2. Dokonaj konwersji wczytanego obrazu z postaci `uint8` na typ `double`

poleceniem `im2double`. Utwórz podaną wcześniej maskę *psf* i wykonaj dekonwolucję z wykorzystaniem funkcji `deconvwnr`. Parametr *Noise-to-signal power ratio* (NSR) uzyskaj na trzy sposoby:

1. $NSR = 1/SNR$
2. $NSR = \text{wariancja_szumu} / \text{wariancja_obrazu}$. Przyjmij wariancję szumu jako $1.0046e-005$. Wariancję obrazu wyznacz z wczytanego obrazu za pomocą składni `var(obraz(:))`, która wyznacza wariancję traktując obraz jako tablicę jednowymiarową.
3. $NSR = 2 * \text{wariancja_obrazu}$
3. Porównaj wyniki odzyskania w jednym oknie dla trzech estymat NSR. Czy udało się odzyskać obraz?
4. Pokaż wyniki prowadzącemu.