 copyright: M. Dwornik
dwornik@agh.edu.pl

Dwuwymiarowa Transformata Fouriera

1 Dwuwymiarowa Szybka Transformata Fouriera

Dwuwymiarowa transformata Fouriera (FFT2) służy przejściu z domeny przestrzeni do domeny częstości. W tym przypadku częstotliwość (częstość) możemy definiować w formie dwóch częstotliwości ortogonalnych (częstotliwość pozioma i pionowa) lub jako częstotliwość wypadkową rozumianą jako odległość od środka obrazu transformaty (po przesunięciu widma ku środkowi).

Dwuwymiarowa Szybka Transformata Fouriera obrazu **I** dana jest wzorem:

$$F(nz, nx) = \beta_L \cdot \sum_{kz=0}^{Nz-1} \left[\sum_{kx=0}^{Nx-1} I(kz, kx) \cdot exp\left(\frac{-2\pi i \cdot kx \cdot nx}{Nx} \right) \right] \cdot exp\left(\frac{-2\pi i \cdot kz \cdot nz}{Nz} \right) \quad (1)$$

Odwrotna transformata Fouriera dla F_{Obrazu} **F** jest dana wzorem:

$$I(kz, kx) = \beta_F \cdot \sum_{nz=0}^{Nz-1} \left[\sum_{nx=0}^{Nx-1} F(nz, nx) \cdot exp\left(\frac{2\pi i \cdot nx \cdot kx}{Nx} \right) \right] \cdot exp\left(\frac{2\pi i \cdot nz \cdot kz}{Nz} \right) \quad (2)$$

gdzie $\beta_L \cdot \beta_F = \frac{1}{M \cdot N}$. Najczęściej $\beta_L = 1$ i $\beta_F = \frac{1}{MN}$. $i = \sqrt{-1}$.

Transformacje te wykonywane są poleceniem: `F=fft2(I)` oraz `I=ifft2(F)` odpowiednio dla prostej i odwrotnej transformaty.

FFT2 wykonywana jest dwuetapowo. Najpierw dokonywany jest szereg transformat jednowymiarowych (np. w wierszach), a następnie dla ortogonalnego wymiaru wykonywane są transformaty na wynikach otrzymanych w poprzednim kroku. Z tego powodu poniższe polecenia MatLABa są równoważne:
`fft2(I)==fft(fft(I).').'`;

2 Widma FFT2

Dla dwuwymiarowej FFT, podobnie jak dla FFT 1D możemy policzyć widma amplitudowe (WA) i fazowe (WF). Dane są one wzorami:

$$WA(m, n) = \sqrt{Im(F(m, n))^2 + Re(F(m, n))^2} \quad (3)$$

$$WF(m, n) = arctg\left(\frac{Im(F(m, n))}{Re(F(m, n))} \right) \quad (4)$$

Do policzenia ich służą polecenia MatLABa:
`WA=abs(F)` oraz `WF=angle(F)`.

Ponieważ energia F skupiona jest na brzegach (składowa zero-częstotliwościowa), niezbędne jest przesunięcie jej do centrum. Służy do tego para poleceń: `FS=fftshift(F)` oraz `F=ifftshift(FS)`.
Uwaga! Polecenia te są wykonywane tylko i wyłącznie na wynikach transformaty oraz nie są symetryczne, tzn. poniższe polecenia nie są równoważne:
`fftshift(fftshift(F))` $\neq$ `ifftshift(fftshift(F))`

copyright: M. Dwornik
dwornik@agh.edu.pl

3 Filtracja w domenie częstotliwości

Głównym zastosowaniem FFT2 jest filtracja obrazów. Polega to na zamianie kosztownej obliczeniowo operacji splotu na znacznie bardziej szybsze mnożenie widm. Podobnie jak w przypadku filtracji sygnałów jednowymiarowych, możemy podzielić filtry na:

- dolnoprzepustowe - przepuszczające dolne częstotliwości, reprezentowane na ogół poprzez trend na obrazie (np. stopniowe zmiany jasności tła);
- górnoprzepustowe - przepuszczające wysokie częstotliwości, reprezentujące na ogół krawędzie, szum i inne krótkookresowe zmiany jasności na obrazie;
- pasmoprzepustowe i pasmowycięciowe - przepuszczające lub wycinające pewien przedział częstotliwości;
- wycięciowe - eliminujące pojedynczą częstotliwość z obrazu (np. efekt pochodzący od sieci elektrycznej).

Filtracja obrazu I przy użyciu filtra M dana jest wzorem:

$$FF = ifft2(|WA(I). * WA(M)| \cdot e^{i(WF(I)+WF(M))}) \tag{5}$$

Wynika z tego rozmiar maski filtru musi być taki sam jak obrazu. Istnieją dwie metody projektowania filtru: bezpośrednio w 2D oraz poprzez rozszerzenie filtrów 1D w 2D.

3.1 Projektowanie filtrów 2D

Idealny, zerofazowy filtr dolnoprzeprzepustowy ma postać centralnie położonego okręgu (o wartościach 1), gdzie długość promienia koła określa pasmo przepuszczania.

Innym przykładem filtru 2D jest dolnoprzepustowy filtr Butterwortha opisany wzorem:

$$H(z,x) = \frac{1}{1 + (D(z,x)/D_0)^{2n}} \tag{6}$$

$$D(z,x) = \sqrt{(x - Nx/2)^2 + (z - Nz/2)^2}, \tag{7}$$

gdzie D_0 odpowiada za częstotliwość odcięcia, n jest rzędem filtru i odpowiada za krzywiznę zbocza filtru, a $[Nz, Nx]$ - rozmiar obrazu.

Filtr Butterwortha w wariancie pasmowozaporowym o szerokości W dany jest wzorem:

$$H(z,x) = \frac{1}{1 + \left(\frac{D(z,x) \cdot W}{D(z,x)^2 - D_0^2}\right)^{2n}} \tag{8}$$

Kolejnym przykładem filtru dolnoprzepustowego jest filtr Gaussowski, dany wzorem:

$$H(z,x) = \exp\left(\frac{-D^2(z,x)}{2 \cdot D_0^2}\right) \tag{9}$$

Do zmiany filtrów 1D w filtry 2D służą następujące polecenia:

- `filtr_1D=remez(N, [częstotliwości], [amplitudy])`, gdzie N jest liczbą współczynników (musi być parzysta). Jako częstotliwości podaje się wartości z przedziału 0-1.
- `filtr_2D=ftrans2(filtr_1D)` - stworzenie osiowo symetrycznego filtru 2D na podstawie filtru 1D.
- `filter2(filtr_2D, obraz)` - filtracja obrazu.

4 Korelacja

Jednym z zastosowań FFT jest funkcja korelacji służąca do wyszukiwania elementów podobnych na obrazie. Do liczenia współczynnika korelacji służy polecenie:
`corr2(obraz1, obraz2);`.
Współczynnik korelacji jest liczony wg następującego wzoru:

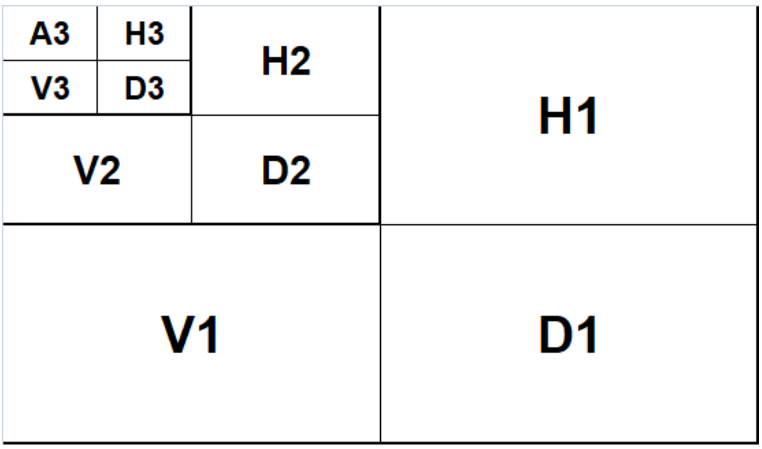
$$r = \frac{\sum_n \sum_m (A_{mn} - \bar{A}) \cdot (B_{mn} - \bar{B})}{\sqrt{(\sum_n \sum_m (A_{mn} - \bar{A})^2) \cdot (\sum_n \sum_m (B_{mn} - \bar{B})^2)}} \tag{10}$$

$\bar{A}$ - wartość średnia obrazu A.

W praktyce sprowadza się to do wyszukiwania maksimów na wyniku polecenie:
`real(ifft2(fft2(obraz).*fft2(rot90(wzorzec, 2), size(obraz))))`

5 Transformata Falkowa 2D

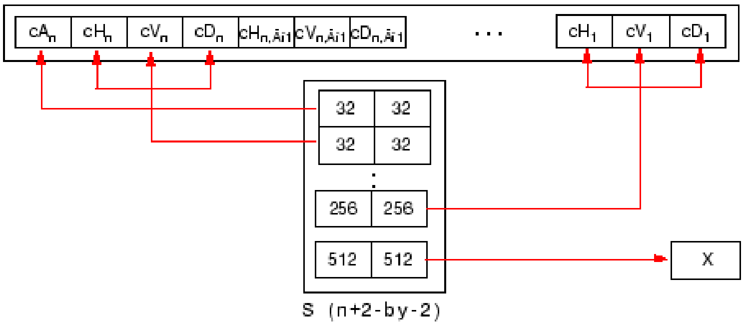
Transformata falkowa jest przykładem analizy czasowo-częstotliwościowej sygnału. Polega na dekompozycji sygnału na część aproksymującą i detale przy użyciu konkretnej falki (wavelet). Odwrotna transformata falkowa polega na rekonstrukcji sygnału bazując na jego składowych. Podobnie jak transformata Fouriera, występuje ona w formie ciągłej (cwt) i dyskretnej (dwt). Transformata falkowa ma zastosowanie w kompresji (np. JPEG 2000), odszumianiu sygnałów czy fuzji obrazów. W przypadku obrazów monochromatycznych dekompozycja falkowa odbywa się z wykorzystaniem funkcji : `[A,H,V,D]=dwt2(X,falka)` dla 1 rzędu i `[C,S]=wavedec2(x, poziom, falka)` dla dowolnego rzędu. W wyniku dekompozycji otrzymujemy 4 obrazy: (A - aproksymacja, H - detale poziome, V - detale pionowe, D - detale ukośne). Rekonstrukcja odbywa się z wykorzystaniem funkcji: `idwt2` oraz `waverec2`.



Rys. 1: Idea dekompozycji falkowej obrazu na 3 poziomie.

Podobnie jak w przypadku sygnałów 1D, tu również dekompozycji na wyższych poziomach podlega tylko i wyłącznie część aproksymacyjna (rys. 1). W wyniku jej działania otrzymujemy submacierze o wymiarze dwukrotnie mniejszym, które są zapisane w wektorze `C` (rys. 2). Do ekstrakcji danych z wektora `C` warto skorzystać z funkcji:
`A=appcoef2(C, S, falka, poziom)` do uzyskania macierzy aproksymacji

i funkcji `detcoef2(opcja, C, S, poziom)` do uzyskania macierzy detali na określonym poziomie. W tym drugim przypadku opcje to `'all'`, jeśli chcemy uzyskać wszystkie 3 macierze detali lub `'h'`, `'v'`, `'d'` dla pojedynczych macierzy.



Rys. 2: Struktura danych wyjściowych w funkcji `wavedec2`. Źródło: MatLAB Wavelet Toolbox.

6 Transformata Gabora

Inną transformatą czasowo-częstotliwościową jest transformata Gabora. Ma ona zastosowanie głównie do detekcji tekstur na obrazach. Polega ona na splocie zbioru kierunkowych masek (tzw. bank filtrów) z obrazem. Maska filtru ma postać zespoloną i stanowi złożenie dwóch elementów: sinusoidalnej fali nośnej 2D o określonej długości fali λ i orientacji θ oraz dwuwymiarowej funkcji Gaussa o zadanym odchyleniu σ, współczynniku eliptyczności γ.

$$G(x, y, \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 \cdot y'^2}{2\sigma^2}\right) \cdot \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \tag{11}$$

gdzie

$$\begin{cases} x' = x \cdot \cos\theta + y \cdot \sin\theta \\ y' = -x \cdot \sin\theta + y \cdot \cos\theta \end{cases} \tag{12}$$

W programie MatLAB transformację Gabora wykonuje się w dwóch krokach:

1. Tworzymy bank filtrów **g** korzystając z polecenia:
   `g = gabor(długość fali, kąty, 'SpatialFrequencyBand', 1.0, 'SpatialAspectRatio', 0.5);`
   Długość fali podaje w pikselach/okres, natomiast kąty (orientacje filtrów) w stopniach. Ilość filtrów w banku jest iloczynem ilości długości fal i ilości orientacji. Dwa ostatnie parametry są opcjonalne i podane z wartościami domyślnymi. Odpowiadają one za częstotliwość fali nośnej (sinusoidalnej) i za rozciągnięcie funkcji Gaussa (stosunek długości osi prostopadłej do kąta orientacji do długości osi zgodnej z kątem orientacji).

2. Filtrując obraz korzystając z polecenia:
   `Magn = imgaborfilt(obraz, bank filtrów);`
   W wyniku filtracji dostajemy zbiór obrazów odpowiadający każdemu z filtrów dostępnemu w banku (najpierw dla pierwszej orientacji wszystkie długości fali, potem dla każdej kolejnej orientacji odpowiadające im długości).