Dyskretne transformacje obrazu

1 Dyskretna Transformacja Cosinusowa (DCT)

Transformacja ta ma zastosowanie przede wszystkim do kompresji obrazów jpg oraz do konwersji mpeg. Para transformat cosinusowych dana jest wzorami:

$$B(p,q) = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} cos\frac{\pi(2m+1)p}{2M} cos\frac{\pi(2n+1)q}{2N} \tag{1}$$

$$A(m,n) = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} cos\frac{\pi(2m+1)p}{2M} cos\frac{\pi(2n+1)q}{2N} \tag{2}$$

gdzie:
$0 \leqslant p \leqslant M-1$, $0 \leqslant q \leqslant N-1$
$0 \leqslant m \leqslant M-1$, $0 \leqslant n \leqslant N-1$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} & dla \quad p = 0 \\ \frac{2}{\sqrt{M}} & dla \quad p \neq 0 \end{cases} \quad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}} & dla \quad q = 0 \\ \frac{2}{\sqrt{N}} & dla \quad q \neq 0 \end{cases}$$

Do implementacji DCT służy polecenie `dct2(obraz, m, n)`, a do transformacji odwrotnej `idct2(obraz, m, n)`. Jeżeli [m,n]<size(obraz), obraz A jest przycinany.

1.1 Kompresja JPEG '92

Kompresja JPEG '92 (ang. *Joint Photographic Experts Group*) oparta jest o transformację DCT2 wykonywaną w oknach o rozmiarze 8x8. Istotą procesu jest kwantyfikacja wyników transformaty oraz późniejsze kodowanie. Proces kompresji wygląda następująco:

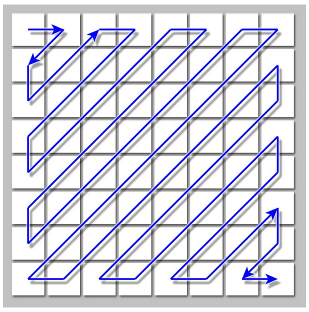
1. Konwersja RGB do YCbCr połączona z przesunięciem średniej do 0.0:

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ Cr &= 128 - 0.168736 \cdot R - 0.331264 \cdot G + 0.53 \cdot B \\ Cb &= 128 + 0.5 \cdot R - 0.418688 \cdot G - 0.081312 \cdot B \end{aligned} \tag{3}$$

   W MatLABie istnieje polecenie konwertujące `rgb2ycbcr()`. Wymagane jest tylko późniejsze, ręczne przesunięcie średniej.
2. Downsampling (redukcja danych)
   - 4:4:4 - Brak downsamplingu
   - 4:2:2 - Redukcja poziomych składowych Cb i Cr o połowę
   - 4:2:0 - Redukcja poziomych i pionowych składowych barwy o połowę.
3. Wykonywanie DCT2 w oknach 8x8.
4. Kwantyfikacja wyników przy wykorzystaniu tablic kodowania Q_Y i Q_C. Polega ona na podzieleniu każdego bloku przez tablicę i zaokrągleniu do najbliższej liczby całkowitej. Dopiero od tego punktu algorytm jest stratny.

$$Q_Y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 69 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

5. Zamiana tablicy na ciąg liczbowy. Wykorzystywana jest ścieżka zygzakowa, by jak najwięcej zer było koło siebie (fig.1).



Rysunek 1: Ścieżka stosowana w kompresji JPG.Źródło *http://en.wikipedia.org/wiki/Zigzag*

6. Kompresja. Stosowany jest najczęściej algorytm Huffmana, choć stosowane też może być kodowanie arytmetyczne (bardziej efektywna kompresja, ale kosztowniejsza obliczeniowo).

2 Transformacja Radona

Transformacja Radona podaje ilość pikseli obrazu binarnego o wartości logicznej prawda w rzucie na prostą umieszczoną pod kątem theta względem obrazu. Transformacja ta dana jest wzorem:

$$R_o(x') = \int_{-\infty}^{\infty} f(x' \cos\theta - y' \sin\theta, x' \sin\theta + y' \cos\theta) dy \tag{4}$$

gdzie:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Do transformacji Radona służy polecenie `[R,xp]=radon(obraz, theta)`, gdzie R - wynik transformacji Radona pod kątem theta (kiedy theta - wektor, R - macierz) xp - wektor zawierający współrzędne kątowe odnoszące się do macierzy R

Do transformacji odwrotnej służy polecenie `iradon(R, theta, parametry)`. Do parametrów zaliczamy:

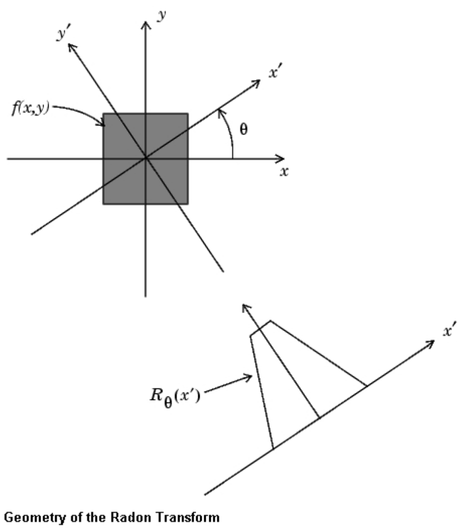
- metoda interpolacji: 'nearest', '**linear**', 'spline'
- metoda filtracji: '**Ram-Lak**', 'Shepp-Logan', 'Cosine', 'Hamming', 'Hann'
- współczynnik skali osi częstotliwości: 0-1, domyślnie 1;
- rozmiar wyjściowy: domyślnie: $n = 2 \cdot floor(size(R, 1)/(2 * \sqrt{(}2)))$

Odwrotna Transformata Radona, w wesji monochromatycznej, jest stosowana w tomografii medycznej do inwersji danych pomiarowych.

3 Transformacja Hougha

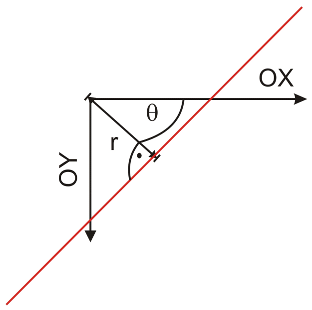
Standardowa Transformacja Hougha (w pakiecie MatLAB dostępna jest tylko dla obrazów logicznych), wykorzystuje parametryczną postać prostej:

$$\varrho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \tag{5}$$



Rysunek 2: Idea Transformacji Radona(Źródło: MATLAB help: Radon transform::description of)

Na fig.3 przedstawiono interpretację geometryczną postaci parametrycznej prostej. Wartość r jest najkrótszą odległością dowolnego punktu prostej od środka układu współrzędnych (czyli długością linii prostopadłej do prostej przechodzącej przez punkt (0,0)), a kąt θ jest kątem pomiędzy dodatnią półosią OX, a linią r.



Rysunek 3: Interpretacja geometryczna parametrycznej postaci prostej

Dzięki temu możemy każdemu punktowi przypisać pewną funkcję sinusoidalną. Linia jest reprezentowana jako punkt przecięcia sinusoid pochodzących od punktów leżących na owej linii. Do implementacji Transformacji Hougha służy polecenie: `[H, theta, rho]=hough(BW, parametry)`. Do parametrów zaliczamy:

- 'ThetaResolution' - Liczba rzeczywista z przedziału 0-90, określająca odległość pomiędzy kolejnymi kątami θ, domyślnie 1;
- 'RhoResolution' - liczba z przedziału: $0 - norm(size(BW))$. Domyślnie 1.

Funkcjami powiązanymi z transformacją Hougha są:

- `peaks = houghpeaks(H, numpeaks, parametry)` - służy ona do lokalizacji punktów przecięć sinusoid na obrazie;
- `houghlines(BW, theta, rho, peaks)` - służy do detekcji linii prostych na obrazie logicznym.