

## PRZETWARZANIE OBRAZÓW CYFROWYCH

## Dyskretna transformata kosinusowa (DCT). Kompresja JPEG.

**Cel:**

- przypomnienie wiadomości o 2D DFT (ang. *Two-Dimensional Discrete Fourier Transform*),
- zapoznanie się z 2D DCT (ang. *Two-Dimensional Discrete Cosine Transform*),
- wprowadzenie do standardu JPEG (ang. *Joint Photographic Experts Group*)
  - kodowanie obrazu w standardzie JPEG,
  - dekodowanie obrazu w standardzie JPEG,
  - kompresja danych

Laboratorium zawiera przedstawienie 2D DCT oraz praktyczną implementację kodera i dekodera dla standardu JPEG z wyłączeniem zapisu danych binarnych w pliku. Szczegółowy opis nagłówek i kolejności przechowywania danych w standardzie JPEG można znaleźć w [6, 8].

**A. Dwuwymiarowa dyskretna transformata kosinusowa**

Dwuwymiarowa dyskretna transformata kosinusowa jest operacją liniową i separowalną. Separowalność oznacza, podobnie jak w przypadku dwuwymiarowej transformaty Fouriera, własność dzięki której wyznaczenie 1D DCT po wierszach (wyrażenie w nawiasie kwadratowym), a następnie kolumnach jest równoznaczne z wyznaczeniem 2D DCT całego obrazu. Obraz wejściowy  $f(m, n)$  jest przekształcany do postaci częstotliwości  $F_{DCT}(u, v)$  poprzez wyrażenie

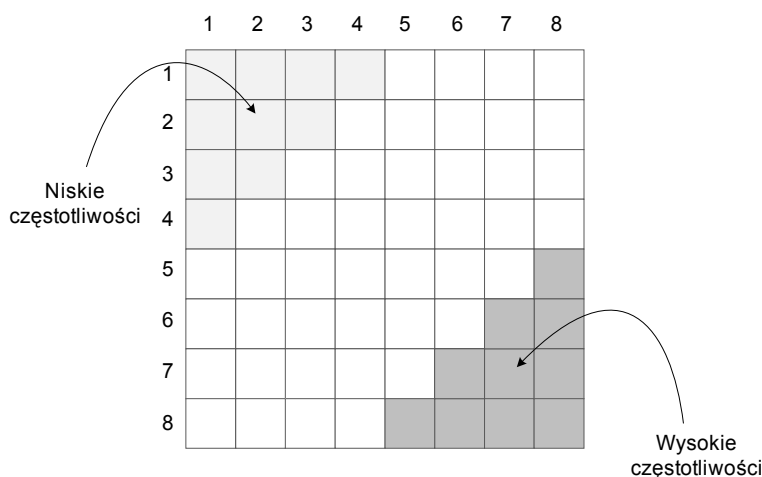
$$F_{DCT}(u, v) = \sum_{m=0}^{M-1} \left[ \sum_{n=0}^{N-1} f(m, n) \cdot \beta(v) \cos\left(\frac{(2n+1)}{2N} \pi v\right) \right] \cdot \alpha(u) \cos\left(\frac{(2m+1)}{2M} \pi u\right)$$

gdzie

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{M}}, & u = 0 \\ \sqrt{\frac{2}{M}}, & u = 1, \dots, M-1 \end{cases} \quad \beta(v) = \begin{cases} \sqrt{\frac{1}{N}}, & v = 0 \\ \sqrt{\frac{2}{N}}, & v = 1, \dots, N-1 \end{cases}$$

Transformata ma za zadanie dokonać dekorrelacji danych, czyli usunąć zależności statystyczne pomiędzy punktami obrazu w dziedzinie przestrzennej. W procesie transformacji dobrany zostaje nowy układ współrzędnych (w oparciu o funkcje bazowe kosinus), gdzie przedstawione dane są niemal minimalnie zależne pomiędzy sobą. W rezultacie tylko kilka współczynników będących rzutami na najważniejsze osie ma istotne wartości.

Przewagą 2D DCT nad 2D DFT jest fakt, że funkcje bazowe 2D DCT mają wartości rzeczywiste w porównaniu do zespolonych 2D DFT. W standardzie JPEG (również MPEG) dyskretna transformata kosinusowa jest wyznaczana dla bloków o wielkości 8x8 (obraz zostaje uprzednio podzielony na bloki 8x8).



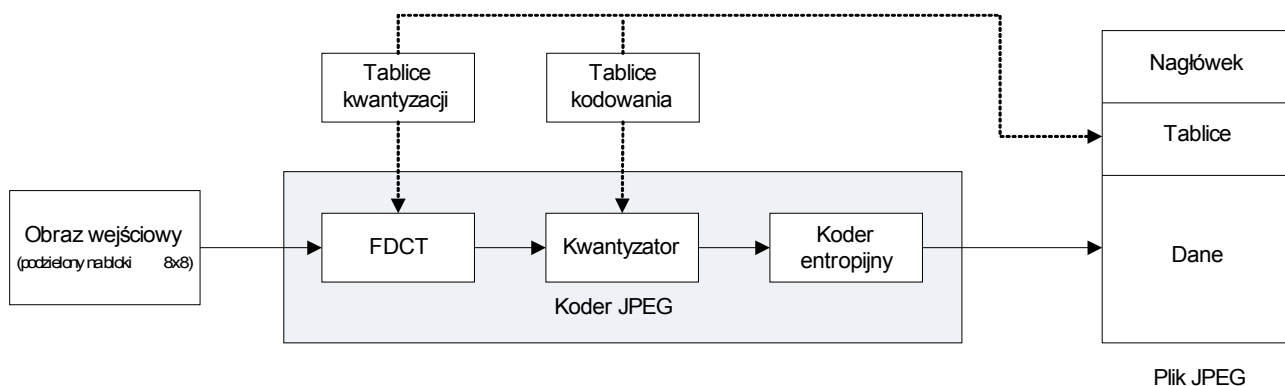
Rysunek 1: Podział częstotliwości w wyniku zastosowanie 2D DCT.

## B. Kodowanie obrazu w standardzie JPEG

Standard JPEG umożliwia trzy alternatywne podejścia kodowania obrazu

1. kompresja stratna, tzw. *sequential baseline system*,
2. niezależna kompresja bezstratna,
3. rozszerzony system kodowania dla systemów wysokiej precyzji

Podczas laboratorium rozważany będzie punkt (1). Ze względu na ograniczony czas ćwiczeń laboratoryjnych zostaną wykonane punkty 1-5 (kodowanie) oraz 4-7 (dekodowanie), z pominięciem operacji *Zig-Zag*, kodowania *RLE* i *Huffmana* oraz zapisu danych do pliku binarnego. Szczegóły na ten temat zawierają książki [6, 8].



Rys 1: Uproszczony schemat blokowy kodera w standardzie JPEG.

Kolejne etapy kodowania obrazu w standardzie JPEG przedstawiają się następująco

1. Przekształcenie modelu barwy RGB obrazu na model YUV zgodnie z zależnością (wykorzystaj funkcję `rgb2ycbcr`)

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

gdzie:

- R, G, B odpowiednio składowa czerwona, zielona, niebieska,
- Y - składowa luminancji
- U - składowa chrominancji (różnica koloru niebieskiego),
- V - składowa chrominancji (różnica koloru czerwonego)

Przekształcenie jest wykonywane, ponieważ oko ludzkie jest znacznie czulsze na zmianę jasności, niż zmianę barwy. Współczynniki kombinacji liniowej dobrane zostały w sposób aposterioryczny (doświadczalny). W zależności od typu wyjścia (monitor) istnieje kilka formuł przekształcenia  $RGB \rightarrow YUV$ .

**Wszystkie kolejne operacje wykonywane są dla każdego kanału Y, U, V.**

2. Podział obrazu wejściowego na bloki 8x8. (do przemyślenia: w jaki sposób należy potraktować obraz w którym jeden, lub obydwa wymiary spełniają zależność  $(Wymiar \bmod 8) \neq 0$  ).
3. Zmiana reprezentacji punktów z *unsigned integer* na *signed integer*. Polega na odjęciu od każdego piksela wartości 128 .
4. Na tak przygotowanych blokach o rozmiarze 8x8 można wykonać dwuwymiarową dyskretną transformatę kosinusową.

Dziedzina częstotliwościowa jest lepszą reprezentacją obrazu, ponieważ istnieje możliwość odseparowania informacji z obrazu, która nie jest istotna z perspektywy percepcji narządu wzroku.

5. Dalszym krokiem jest wykonanie na każdym z bloków 8x8 operacji kwantyzacji zdefiniowanej jako

$$Z(m, n) = \text{Round} \left( \frac{F_{DCT}(m, n)}{c \cdot Q(m, n)} \right)$$

gdzie

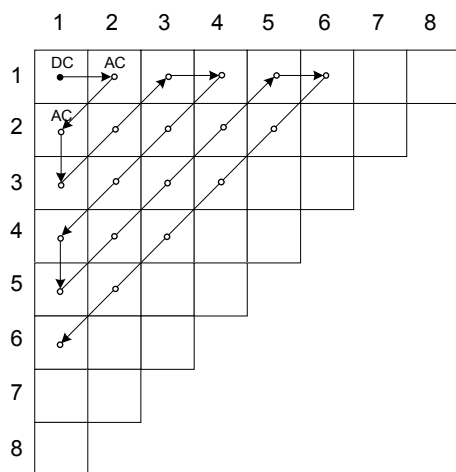
- $F_{DCT}(m, n)$  – F-obraz obrazu  $f(m, n)$
- $Q(m, n)$  – tablica kwantyzacji dla danego kanału Y, U, V
- $c$  - współczynnik kompresji obrazu (standardowo  $c=1$  )

Dla  $c > 1$  występuje zwiększenie kompresji (pogorszenie jakości). Dla  $c \in (0, 1)$  zmniejszenie kompresji (polepszenie jakości). Nie istnieje natomiast możliwość polepszenia jakości obrazu już skompresowanego.

Kwantyzacja składowej Y odbywa się w oparciu o tablicę kwantyzacji  $Q_Y$ , natomiast kwantyzacja składowych U oraz V w oparciu o tablicę kwantyzacji  $Q_C$  (plik `tablice_kwantyzacji.mat`).

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

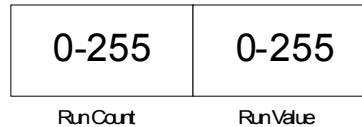
6. Kolejnym krokiem jest wykonanie na każdym z bloków  $8 \times 8$  operacji *Zig-Zag*. Polega ona na reorganizacji bloku w sposób pokazany na rysunku, wynikiem czego jest wektor 64 elementowy. W rezultacie tablica  $8 \times 8$  zostaje przekształcona w wektor.



Rys 2: Schemat reorganizacji danych zgodnie z operacją Zig-Zag

7. Każdy 64-elementowy wektor wynikowy należy zakodować za pomocą *kodowania RLE* (ang. *Run Length-Encoding*).

Kodowanie RLE opiera się na pojęciu atomu, czyli niepodzielnej jednostce danych. W przypadku kodowania punktów obrazu tą jednostką jest bajt.

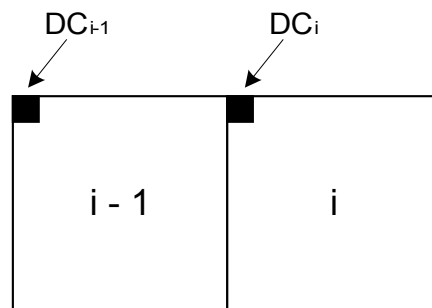


Kolejne wartości kodowane są w postaci dwóch bajtów. Pierwszy z nich oznacza (*ilość powtórzeń -1*), drugi *wartość punktu*.

Przykładowy wektor po operacji Zig-Zag  $[100, 0, 0, 0, 0, 0, 0, \dots, 0]_{1 \times 64}$  zostanie zakodowany w postaci  $[0, 100, 62, 0]$ , co oznacza jedno powtórzenie wartości 100 oraz 63 powtórzeń wartości 0.

8. Ostatnim etapem jest kodowanie binarne za pomocą *algorytmu Huffmana* oraz zapis do pliku binarnego zgodnie ze specyfikacją.

W specyfikacji formatu JPEG składowe AC i DC są kodowane oddzielnie. Zapis składowej DC w  $i$ -tym bloku jest zastępowany różnicą w stosunku do składowej DC w bloku  $i-1$ .



Więcej informacji na ten temat zawiera podręcznik [5].

### C. Dekodowanie obrazu w standardzie JPEG

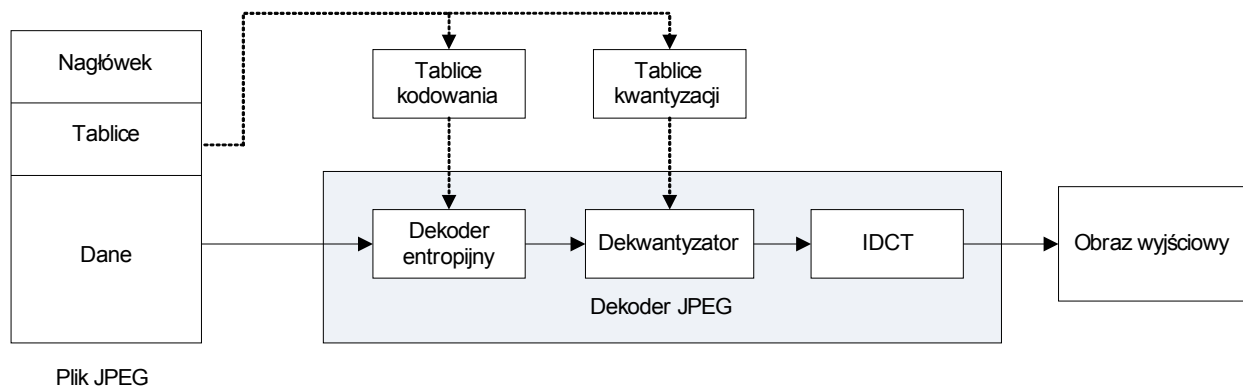
Odwrotna dyskretna transformata kosinusowa dla F-obrazu wejściowego  $F_{DCT}(u, v)$  jest określona poprzez wyrażenie

$$x(m, n) = \sum_{v=0}^{N-1} \left[ \sum_{u=0}^{M-1} F_{DCT}(u, v) \cdot \alpha(u) \cos\left(\frac{(2m+1)}{2M} \pi u\right) \right] \cdot \beta(v) \cos\left(\frac{(2n+1)}{2N} \pi v\right)$$

gdzie

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{M}}, & u = 0 \\ \sqrt{\frac{2}{M}}, & u = 1, \dots, M-1 \end{cases} \quad \beta(v) = \begin{cases} \sqrt{\frac{1}{N}}, & v = 0 \\ \sqrt{\frac{2}{N}}, & v = 1, \dots, N-1 \end{cases}$$

W przypadku dekodera obrazu JPEG wykorzystywana jest dla bloków o rozmiarze 8x8.



Rys 3: Uproszczony schemat blokowy dekodera w standardzie JPEG.

Kolejne etapy dekodowania obrazu w standardzie JPEG przedstawiają się następująco:

1. Dekompresja danych za pomocą algorytmu Huffmana
2. Dekodowanie danych za pomocą algorytmu RLE.
3. Utworzenie bloków o rozmiarze 8x8 dla każdego z kanałów Y, U, V oraz wstawienie wartości wektorów do kolejnych macierzy metodą Zig-Zag począwszy od lewego górnego rogu, do prawego dolnego.

4. Wykonanie na każdym z bloków 8x8 operacji dekwantyzacji zdefiniowanej jako

$$F_{DCT}(m, n) = Z(m, n) \cdot c \cdot Q(m, n)$$

gdzie

- $Z(m, n)$  – skwantowane wartości F-obrazu  $f(m, n)$
- $Q(m, n)$  – tablica kwantyzacji dla danego kanału Y, U, V
- $c$  - współczynnik kompresji obrazu (standardowo  $c=1$  )

**Uwaga!**

Tablice  $Q(m, n)$  muszą być takie same jak podczas kodowania. Standard JPEG umożliwia podanie własnych tablic kwantyzacji, pod warunkiem ich zapamiętania. Inaczej dekompresja nie będzie możliwa albo będzie zawierała błędy. W przypadku kodowania stratnego należy zapamiętać również współczynnik  $c$ .

Dekwantyzacja składowej Y odbywa się w oparciu o tablicę kwantyzacji  $Q_Y$ , natomiast dekwantyzacja składowych U oraz V w oparciu o tablicę kwantyzacji  $Q_C$  (plik `tablice_kwantyzacji.mat`).

5. Na tak przygotowanych blokach o rozmiarze 8x8 można wykonać dwuwymiarową odwrotną dyskretną transformatę kosinusową.
6. Połączenie wyników 2D IDCT dla bloków 8x8 w całość dla każdego z kanałów Y, U, V.
7. Przekształcenie modelu barwy YUV obrazu na model RGB zgodnie z zależnością (funkcja `rgb2ycbcr`).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 2.0257 & 1.8175 & 1.4018 \\ 0.4775 & -1.2699 & -0.714 \\ 0.9999 & 1.7718 & -0.0001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

gdzie:

- R, G, B odpowiednio składowa czerwona, zielona, niebieska,
- Y - składowa luminancji
- U - składowa chrominancji (różnica koloru niebieskiego),
- V - składowa chrominancji (różnica koloru czerwonego)

## Literatura

1. Wallace G. K., *The JPEG Still Picture Compression Standard*, IEEE Transactions on Consumer Electronics, December 1991, <http://white.stanford.edu/~brian/psy221/reader/Wallace.JPEG.pdf>
2. Gonzalez R. C., Woods R. E. *Digital Image Processing*, Pearson Education, New Jersey 2008,
3. Gonzalez R. C., Woods R. E., Eddins S. L. *Digital Image Processing Using Matlab*, Prentice Hall Education, 2004,
4. Zieliński T. P. *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań.*, Wydawnictwa Komunikacji i Łączności, Warszawa 2005,
5. Drozdek A., *Wprowadzenie do kompresji danych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2007,
6. Skarbek W. et al., *Multimedia. Algorytmy i standardy kompresji*, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998
7. Przelaskowski A., *Kompresja danych. Podstawy, Metody bezstratne, Kodery obrazów*, Wydawnictwo BTC, Warszawa 2005,
8. Levine J. *Programowanie plików graficznych w C/C++*, Wydawnictwo Translator s.c., Warszawa 1994,
9. Stranneby D., *Cyfrowe przetwarzanie sygnałów. Metody, Algorytmy, Zastosowania.*, Wydawnictwo BTC, Warszawa 2005,
10. Heim K. *Metody kompresji danych*, MIKOM, Warszawa 2000