

# Techniki multimedialne

## Ćwiczenie 4

### Kompresja i dekompresja obrazów - JPEG

---

Opracował Dawid Warchoł  
Politechnika Rzeszowska  
Katedra Informatyki i Automatyki

#### Cel ćwiczenia

Ćwiczenie to ma na celu zapoznanie studenta z algorytmem kompresji obrazów JPEG, a w szczególności z następującymi jego elementami:

- podział obrazu na bloki i składanie obrazu z bloków,
- dyskretna transformata kosinusowa (DCT) i odwrotna dyskretna transformata kosinusowa (IDCT),
- kwantyzacja i dekwantyzacja.

Należy napisać program realizujący kompresję oraz dekompresję JPEG z pominięciem kodowania Zig-Zag oraz kodowania Huffmana. Program należy napisać w języku C++ z wykorzystaniem biblioteki OpenCV oraz wstępnie przygotowanych fragmentów kodu zapisanych w pliku *jpeg\_todo.cpp*. Fragmenty te powinny być uzupełnione zgodnie z komentarzami umieszczonymi w odpowiednich miejscach oraz instrukcją wykonania ćwiczenia przedstawioną poniżej. Plik z kodem programu oraz dodatkowe pliki (np. obrazów) zapewnia prowadzący zajęcia.

Wskazówka: Dyskretna transformata kosinusowa jest transformacją całkowicie odwracalną (poprzez IDCT). Dopiero w procesie kwantyzacji/dekwantyzacji część informacji zostaje tracona. Aby zaobserwować efekt każdego etapu algorytmu, należy wykonać poniższe zadania w następującej kolejności: 1, 5, 2, 4, 3, 6. Po wykonaniu kroku 4 warto sprawdzić, czy obraz uzyskany po złożeniu bloków wynikowych odwrotnej dyskretniej transformaty kosinusowej jest identyczny z oryginalnym. Jeśli tak, to znaczy, że ćwiczenie do tej pory zostało wykonane poprawnie i można zrobić kwantyzację/dekwantyzację oraz przeanalizować obraz wyjściowy.

Uwaga: Nie należy wpisywać wymiarów poszczególnych obrazów "na sztywno" w kodzie. Warto skorzystać z pól `rows` i `cols` obiektów przechowujących obraz.

## Zadanie 1. Podział obrazu na bloki

- a) Wczytać do pamięci obraz JPEG, którego wysokość oraz szerokość są podzielne przez 8 (domyślnie plik *fruits2.bmp*). Obraz powinien być przechowywany jako 1-kanałowa (szara) mapa bitowa.

Przetwarzamy szary obraz dla przyspieszenia obliczeń i uproszczenia kodu. Implementacja algorytmu JPEG na obrazie kolorowym wymagałaby wczytania 3-kanałowej mapy bitowej i wykonania wszystkich kroków dla trzech osobnych kanałów.

- b) Podzielić obraz na bloki o rozmiarze 8x8 pikseli i zapisać je do tablicy dwuwymiarowej tablicy bloków.

## Zadanie 2. Dyskretna transformata kosinusowa

Wykonać operację dyskretną transformaty kosinusowej na blokach utworzonych w poprzednim punkcie. DCT jest przekształceniem, którego wynikiem jest blok o takich samych rozmiarach jak blok wejściowy. Składa się on z jednego współczynnika DC (element w lewym górnym rogu) oraz wielu współczynników AC (elementy pozostałe). Współczynnik DC jest średnią wartości wszystkich pikseli danego bloku, natomiast współczynniki AC reprezentują zmiany wartości pikseli w całym bloku. Warto zauważyć, że współczynnik DC jest w naszym przypadku liczbą dodatnią z przedziału [0-255], natomiast współczynniki AC mogą być liczbami dodatnimi lub ujemnymi, przy czym ich wartości są bardziej zbliżone do zera dla elementów znajdujących się dalej od lewego górnego rogu.

DCT należy wykonać dla wszystkich pikseli każdego z bloków według następującego wzoru:

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

gdzie:

- $F[u, v]$  - rezultat DCT dla danego piksela;
- $N$  - rozmiar bloku (wysokość lub szerokość bloku w pikselach);
- $u, v$  - współrzędne piksela, dla którego liczona jest DCT;
- $f[m, n]$  - piksel o współrzędnych  $m, n$  bloku wejściowego.

## Zadanie 3. Kwantyzacja i dekwantyzacja

- a) Dokonaj kwantyzacji bloków wynikowych uzyskanych w poprzednim zadaniu. Operacja ta powoduje utratę informacji obrazu i polega na podzieleniu wszystkich elementów bloku przez odpowiadające im elementy zmodyfikowanej macierzy kwantyzacji. Po wykonaniu dzielenia wynik należy zaokrąglić do najbliższej liczby całkowitej. Macierz kwantyzacji,

którą przedstawiono poniżej, powinna zostać wprowadzona do dwuwymiarowej tablicy liczb zmiennoprzecinkowych.

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

Po wprowadzeniu macierzy należy ją zmodyfikować w następujący sposób:

$$qMatrix = qMatrix \cdot \frac{100 - qualityLevel}{99} \cdot 0.18 + 0.02,$$

gdzie *qualityLevel* oznacza wskaźnik jakości kompresowanego obrazu i powinien przyjmować wartości z zakresu 1-100. Im większy wskaźnik jakości, tym mniejszy stopień kompresji.

Powyższa operacja powoduje mnożenie elementów macierzy przez wartość oznaczającą stopień jakości przeskalowany z zakresu 1-100 do 0.2-0.02.

- b) Dokonaj dekwantyzacji bloków uzyskanych w poprzednim kroku. Operacja ta polega na przemnożeniu wszystkich elementów bloku przez odpowiadające im elementy zmodyfikowanej macierzy kwantyzacji.

Zaobserwuj na czym polega utrata informacji w procesie kwantyzacji/dekwantyzacji. Czy pomijając zaokrąglanie do najbliższych liczb całkowitych (kwantyzacja) informacja zostałaby utracona?

## Zadanie 4. Odwrotna dyskretna transformata kosinusowa

Odwrotna dyskretna transformata kosinusowa jest transformacją odwracającą efekty DCT. Za jej pomocą możemy uzyskać obraz identyczny z oryginalnym. IDCT należy jednak wykonać na blokach uzyskanych po kwantyzacji/dekwantyzacji, co spowoduje uzyskanie obrazu o nieco gorszej jakości od oryginalnego.

IDCT należy wykonać dla wszystkich pikseli każdego z bloków według następującego wzoru:

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

gdzie:

- $f[m, n]$  - rezultat IDCT dla danego piksela;
- $N$  - rozmiar bloku (wysokość lub szerokość bloku w pikselach);
- $m, n$  - współrzędne piksela, dla którego liczona jest IDCT;
- $F[u, v]$  - piksel o współrzędnych  $u, v$  bloku wejściowego (bloku wynikowego DCT);
- $c[\lambda] = 1$  dla  $\lambda = 0$  oraz  $c[\lambda] = 2$  dla  $\lambda = 1, 2, 3, \dots, N-1$ .

## Zadanie 5. Składanie obrazu z bloków wynikowych

Piksele bloków wynikowych otrzymanych po wykonaniu odwrotnej dyskretnej transformaty kosinusowej należy skopiować do zmiennej przechowującej obraz. Ostatnim krokiem jest wyświetlenie obrazu.

Przed wykonaniem wyżej opisanej operacji należy upewnić się, czy bloki wynikowe IDCT nie zawierają elementów mniejszych niż 0 oraz większych niż 255. Takie elementy należy zastąpić wartościami odpowiednio 0 i 255.

## Zadanie 6. Obserwacja zmiany jakości kompresji w zależności od stopnia kompresji

Wykonaj napisany program dla różnych wartości zmiennej *qualityLevel*. Zaobserwuj jak pogarsza się jakość obrazu dla mniejszych wartości zmiennej. Na czym polega utrata jakości w stosunku do obrazu oryginalnego?

## Źródła

- [1] W. B. Pennebaker, J. L. Mitchell: *JPEG still image data compression standard (3rd ed.)*. Springer, 1993.
- [2] A. Tyagi: *The DCT/IDCT Solution Customer Tutorial* (<http://www.slideshare.net/chetanrao2012/discrete-cosine-transform>), 2000.
- [3] OpenCV 3.4.5. dokumentacja: <https://docs.opencv.org/3.4.5/>.