
Inżyniera Obrazów

laboratorium numer 3

Autor sprawozdania: Michał Dziedziak 263901

Imię i nazwisko prowadzącego kurs: dr inż. Jan Nikodem

Dzień i godzina zajęć: czwartek, 11:15 - 14:15

Spis treści

1	Temat laboratorium	2
2	Zadania do wykonania i plan pracy	2
2.1	Zadanie 4: Implementacja części algorytmu JPEG	2
2.2	Zadanie 5: Dokończenie implementacji algorytmu JPEG	2
3	Teoria	2
3.1	Konwersja modelu barw RGB do YCbCr	3
3.2	Przeskalowanie w dół (downsampling) macierzy składowych Cb i Cr	3
3.3	Podział obrazu na bloki o rozmiarze 8x8	3
3.4	Wykonanie dyskretnej transformacji kosinusowej (DCT)	3
3.5	Podzielenie każdego bloku obrazu przez macierz kwantyzacji	4
3.6	Zaokrąglenie wartości w każdym bloku do liczb całkowitych	4
3.7	Zwinięcie każdego bloku 8x8 do wiersza 1x64 - algorytm ZigZag	4
3.8	Zakodowanie danych obrazu	4
4	Prezentacja wykonanego zadania	5
5	Wnioski	6

Spis rysunków

1	Logi konsoli opisujące rozmiary obrazów po kompresji (z różnymi parametrami)	5
2	Prezentacja oryginalnego obrazu wraz z wariacjami po kompresji i dekompresji (z różnymi parametrami)	5
3	Logi konsoli prezentujące współczynniki MSE pomiędzy oryginalnym obrazem a obrazem po kompresji i dekompresji (z różnymi parametrami)	6

1 Temat laboratorium

W ramach trzecich zajęć laboratoryjnych mieliśmy wykonać zadanie 4 i 5 z listy drugiej. Punkty te dotyczyły implementacji algorytmu JPEG.

2 Zadania do wykonania i plan pracy

2.1 Zadanie 4: Implementacja części algorytmu JPEG

Zadanie czwarte zakładało uproszczoną implementację algorytmu JPEG. W ramach tego punktu mieliśmy zaimplementować:

- Kroki: 0, 1, 2, 3, 7, 8 algorytmu JPEG. Gdzie:
 - Krok 0: Wczytanie obrazu wejściowego,
 - Krok 1: Konwersja modelu barw: RGB \rightarrow YCbCr,
 - Krok 2: Przeskalowanie w dół macierzy składowych Cb i Cr,
 - Krok 3: Podział obrazu na bloki o rozmiarze 8x8,
 - Krok 7: Zwinięcie każdego bloku 8x8 do wiersza 1x64 - algorytm ZigZag,
 - Krok 8: Zakodowanie danych obrazu,
- Zmierzyć liczbę bajtów powstałego obrazu po kroku 8
- Ocenąć wpływ kroku 2-go na rozmiar i wygląd, poprzez stworzenie trzech wariacji obrazu:
 - bez próbkowania,
 - z próbkowaniem co drugi element,
 - z próbkowaniem co czwarty element.
- Dokonać dekompresji poprzez odwrócenie powyższych kroków.

2.2 Zadanie 5: Dokończenie implementacji algorytmu JPEG

Zadanie piąte zakładało dokończenie implementacji algorytmu JPEG. W ramach tego punktu mieliśmy zaimplementować:

- Pozostałe kroki algorytmu:
 - Krok 4: Wykonanie dyskretnej transformacji cosinusowej na każdym bloku obrazu,
 - Krok 5: Podzielenie każdego bloku obrazu przez macierz kwantyzacji,
 - Krok 6: Zaokrąglenie wartości w każdym bloku do liczb całkowitych.
- Ocenąć jak wybór czynnika QF (quality factor) wpływa na rozmiar i wygląd obrazka.
- Dokonać dekompresji poprzez odwrócenie powyższych kroków.

3 Teoria

Format JPEG (ang. Joint Photographic Experts Group) jest powszechnie stosowanym standardem kompresji stratnej do obrazów cyfrowych. Algorytm JPEG składa się z kilku etapów, które pozwalają na efektywną redukcję rozmiaru pliku przy minimalnej utracie jakości obrazu.

Algorytm JPEG jest symetryczny, co oznacza, że dekompresja jest procesem odwrotnym do kompresji (dokonuje jej się poprzez odwrócenie kolejności kroków kompresji).

3.1 Konwersja modelu barw RGB do YCbCr

W pierwszym kroku obraz jest konwertowany z modelu barw RGB do przestrzeni barw YCbCr. Model YCbCr składa się z trzech składowych: Y (luminancja), Cb (chrominancja niebieska) i Cr (chrominancja czerwona). Konwersja ta pozwala na oddzielenie informacji o jasności od informacji o barwie, co ułatwia dalszą kompresję.

3.2 Przeskalowanie w dół (downsampling) macierzy składowych Cb i Cr

Kolejnym krokiem jest przeskalowanie macierzy składowych Cb i Cr, które odpowiadają za informacje o barwie, przy użyciu technik próbkowania, takich jak próbkowanie co drugi element lub co czwarty element. To przeskalowanie pozwala na zmniejszenie rozmiaru danych.

3.3 Podział obrazu na bloki o rozmiarze 8x8

Obraz jest dzielony na niewielkie bloki o stałym rozmiarze 8 na 8 pikseli. Ten krok umożliwia stosowanie transformacji i kwantyzacji na mniejszych fragmentach obrazu, co ułatwia kompresję.

3.4 Wykonanie dyskretnej transformacji kosinusowej (DCT)

Każdy blok pikseli jest poddawany dyskretnej transformacji kosinusowej. DCT zamienia blok pikseli na zestaw współczynników reprezentujących jego składowe częstotliwości. Współczynniki te są obliczane za pomocą następującego równania:

$$F(u, v) = \frac{2}{\sqrt{MN}} C(u) C(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cos\left(\frac{(2m+1)u\pi}{2M}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right) \quad (1)$$

gdzie:

- $C(u, v) = \frac{1}{\sqrt{2}}$ dla $u, v = 0$, w przeciwnym wypadku $C(u, v) = 1$
- M - ilość wierszy w wejściowym zestawie danych
- N - ilość kolumn w wejściowym zestawie danych
- m - indeksy próbek z wiersza w domenie przestrzeni, gdzie $0 \leq m \leq M - 1$
- n - indeksy próbek z kolumny w domenie przestrzeni, gdzie $0 \leq n \leq N - 1$
- $f(m, n)$ - wartości przestrzenne w bloku danych wejściowych
- u - indeks wiersza w domenie częstotliwości
- v - indeks kolumny w domenie częstotliwości
- $F(u, v)$ - współczynniki transformaty w domenie częstotliwości

Przy założeniu, że $M = N = 8$ równanie przyjmuje postać:

$$F(u, v) = \frac{C(v)}{2} \sum_{n=0}^7 \left[\frac{C(u)}{2} \sum_{m=0}^7 f(m, n) \cos\left(\frac{(2m+1)u\pi}{16}\right) \right] \cos\left(\frac{(2n+1)v\pi}{16}\right) \quad (2)$$

Transformacja DCT pozwala skutecznie kompresować obrazy poprzez selektywne wyodrębnianie istotnych cech, takich jak krawędzie i tekstury, oraz eliminowanie *szumów* niewidocznych dla ludzkiego oka. Dzięki analizie częstotliwościowej, DCT umożliwia minimalizację rozmiaru pliku przy zachowaniu jakości obrazu.

3.5 Podzielenie każdego bloku obrazu przez macierz kwantyzacji

Po DCT dla każdego bloku pikseli, współczynniki są dzielone przez odpowiadające im elementy macierzy kwantyzacji.

Macierz kwantyzacji zawiera wartości, które określają, jak mocno każdy współczynnik DCT zostanie zaokrąglony. Im większe wartości w macierzy kwantyzacji, tym większa kompresja danej informacji.

Parametr jakości (**quality factor**) jest używany do kontrolowania stopnia kompresji i jakości obrazu. Zwiększenie wartości tego parametru prowadzi do mniejszego kompresowania współczynników DCT, co oznacza mniejszą stratę informacji i lepszą jakość obrazu, ale także większy rozmiar pliku. Zmniejszenie wartości quality factor powoduje większą kompresję danych, co prowadzi do mniejszego rozmiaru pliku kosztem jakości obrazu.

W tym kroku algorytm staje się stratny.

Macierze kwantyzacji, które zostały użyte w zadaniu:

$$\text{Macierz kwantyzacji dla kanału Y} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 48 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad (3)$$

$$\text{Macierz kwantyzacji dla kanałów Cb i Cr} = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \quad (4)$$

3.6 Zaokrąglenie wartości w każdym bloku do liczb całkowitych

Po podzieleniu przez macierz kwantyzacji wartości wynikowe są zaokrąglane do najbliższych liczb całkowitych, co dodatkowo redukuje rozmiar danych.

3.7 Zwinięcie każdego bloku 8x8 do wiersza 1x64 - algorytm ZigZag

Następnie dokonuje się transformacji macierzy 8x8, do jednego ciągu 64 elementów. Proces ten jest realizowany przy użyciu algorytmu ZigZag. Algorytm ten polega na skanowaniu bloku 8x8 w sposób skośny, przemieszczając się między kolejnymi elementami macierzy wzdłuż ścieżki przypominającej kształt litery "Z". W efekcie, kolejne wartości są uporządkowane w jednym ciągu, który jest następnie wykorzystywany do dalszej obróbki i kodowania danych.

3.8 Zakodowanie danych obrazu

W ostatnim etapie procesu kompresji obrazu, wykorzystuje się techniki kodowania danych w celu dalszej redukcji rozmiaru pliku. Wykorzystałem w tym celu algorytm DEFLATE, który jest popularnym i wydajnym algorytmem kompresji danych.

Algorytm DEFLATE łączy w sobie dwa główne mechanizmy kompresji: algorytm LZ77 (Lempel-Ziv 77) oraz algorytm Huffmana.

Pierwszym krokiem jest zastosowanie algorytmu LZ77, który polega na wyszukiwaniu powtarzających się sekwencji danych w strumieniu wejściowym i ich zastępowaniu krótszymi symbolami. Dzięki temu możliwe jest skuteczne redukcowanie ilości danych w strumieniu obrazu.

Następnie dane poddawane są procesowi kodowania przy użyciu algorytmu Huffmana. W tym kroku, często występujące symbole są mapowane na krótsze kody binarne, podczas gdy rzadziej występujące symbole są mapowane na dłuższe kody binarne. To pozwala na dodatkową redukcję rozmiaru danych poprzez przypisanie krótszych kodów binarnych do często występujących symboli.

Dzięki zastosowaniu algorytmu DEFLATE, możliwe jest osiągnięcie wysokiego stopnia kompresji danych obrazu bez znacznego pogorszenia jakości obrazu.

Źródła: Fileformat [2019], inż. Zbigniew Małach [2009]

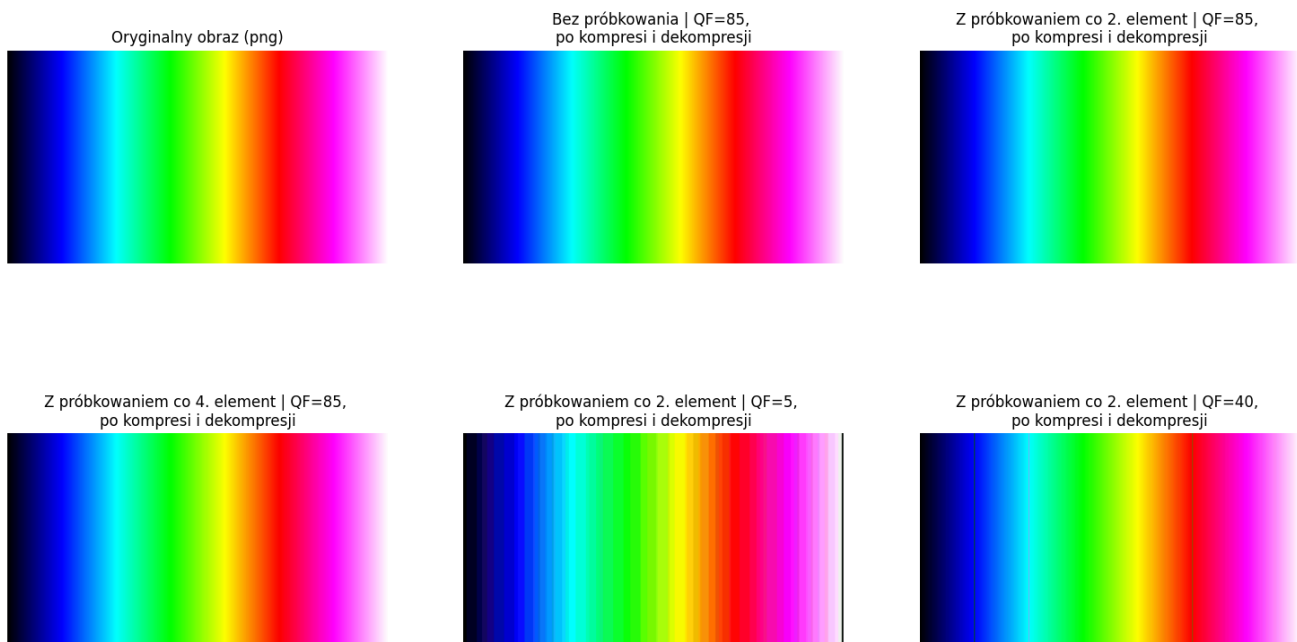
4 Prezentacja wykonanego zadania

Rozmiary obrazu po kompresji (z różnymi parametrami)

```
[ROZMIAR]
Rozmiar dla: bez próbkowania, QF=85: 182.92 KB
Rozmiar dla: próbkowanie co 2, QF=85: 97.52 KB
Rozmiar dla: próbkowanie co 4, QF=85: 73.33 KB
Rozmiar dla: próbkowanie co 2, QF=5: 83.33 KB
Rozmiar dla: próbkowanie co 2, QF=40: 96.84 KB
```

Rysunek 1: Logi konsoli opisujące rozmiary obrazów po kompresji (z różnymi parametrami)

Obraz po kompresji i dekompresji (z różnymi parametrami)



Rysunek 2: Prezentacja oryginalnego obrazu wraz z wariacjami po kompresji i dekompresji (z różnymi parametrami)

Wartości współczynnika MSE pomiędzy oryginalnym obrazem a obrazem po kompresji i dekompresji (z różnymi parametrami)

```
[MSE]
MSE pomiędzy oryginałem, a obrazem bez próbkowania i QF=85: 1.13
MSE pomiędzy oryginałem, a obrazem z próbkowaniem co 2. element i QF=85: 1.34
MSE pomiędzy oryginałem, a obrazem z próbkowaniem co 4. element i QF=85: 2.2
MSE pomiędzy oryginałem, a obrazem z próbkowaniem co 2. element i QF=5: 38.68
MSE pomiędzy oryginałem, a obrazem z próbkowaniem co 2. element i QF=40: 3.82
```

Rysunek 3: Logi konsoli prezentujące współczynniki MSE pomiędzy oryginalnym obrazem a obrazem po kompresji i dekompresji (z różnymi parametrami)

5 Wnioski

Podczas laboratorium pomyślnie zaimplementowałem algorytm JPEG. Na podstawie wykonanych zadań zauważyłem, że wybór parametrów takich jak czynnik jakości (QF) czy technika próbkowania ma znaczący wpływ na rozmiar i jakość obrazu.

- Wraz ze wzrostem czynnika jakości (QF) rośnie jakość obrazu, ale maleje stopień kompresji (co za tym idzie, zwiększa się rozmiar obrazu).
- Technika próbkowania również wpływa na rozmiar obrazu. Wraz ze wzrostem stopnia próbkowania maleje rozmiar obrazu, ale rośnie utrata jakości.

Literatura

Fileformat. Jpeg - image file format. <https://docs.fileformat.com/image/jpeg/>, 2019.

inż. Zbigniew Małach. Kompresja jpeg. https://gimpuj.info/zbyma/Poradnik-Kompresja_JPEG.pdf, 2009.