

Analiza sygnału w dziedzinie czasu i częstotliwości

Ćwiczenie 6: Przetwarzanie obrazów. Format graficzny JPEG i JPEG 2000

1. Wstęp

Celem ćwiczenia jest zapoznanie się z zasadą działania kompresji obrazów w formacie JPEG oraz JPEG 2000 oraz przeprowadzenie transformacji DCT na obrazie szaro-skalowym.

Użyte oprogramowanie: *Python ver. 3.9.7*

Użyte biblioteki: *numpy, scipy, matplotlib, imageio*

2. Kod źródłowy

```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
import matplotlib.pyplot as pylab
import imageio
import scipy.fftpack as fftpack
from numpy import pi
from numpy import sin
from numpy import zeros
from numpy import r_
from scipy import signal
from scipy import misc

plt.rcParams["figure.figsize"] = [15, 8]
plt.rcParams['font.size'] = '13'
pylab.rcParams['figure.figsize'] = (15, 8)

im = imageio.imread("semestr2\ASwDCiC\lab6\lena.pgm").astype(float)

plt.imshow(im, cmap='gray')
```

```

plt.show()

def dct2(a):
    return fftpack.dct(fftpack.dct( a, axis=0, norm='ortho' ), axis=1, norm='ortho' )

def idct2(a):
    return fftpack.idct(fftpack.idct( a, axis=0 , norm='ortho'), axis=1 , norm='ortho')

def do_dct(img, n):
    imsize = img.shape
    dct = np.zeros(imsize)

    for i in range(0, imsize[0], n):
        for j in range(0, imsize[1], n):
            dct[i:(i+n), j:(j+n)] = dct2( img[i:(i+n), j:(j+n)] )

    return dct

def undo_dct(dct, n, imsize):

    im_dct = np.zeros(imsize)

    for i in range(0, imsize[0], n):
        for j in range(0, imsize[1], n):
            im_dct[i:(i+n), j:(j+n)] = idct2(dct[i:(i+n), j:(j+n)] )

    return im_dct

n = 8

imsize = im.shape
ddct = do_dct(im, n)

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('8x8 DCT bez kwantyzacji oraz transformacja odwrotna')
ax1.imshow(ddct, cmap='gray', vmax = np.max(ddct)*0.01, vmin = 0)
ax2.imshow(undo_dct(ddct, n, imsize) , cmap='gray')
plt.show()

thresh = 0.01
dct_thresh = ddct * (abs(ddct) >= (thresh*np.max(ddct)))
percent_nonzeros = np.sum( dct_thresh != 0.0 ) / dct_thresh.shape[0]**2

print (f"Keeping only {percent_nonzeros} of the DCT coefficients")

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('8x8 DCT z kwantyzacją oraz transformacja odwrotna')
ax1.imshow(dct_thresh, cmap='gray', vmax = np.max(dct_thresh)*0.01, vmin = 0)
ax2.imshow(undo_dct(dct_thresh, n, imsize) , cmap='gray')

```

```

plt.show()

n = 16

imsize = im.shape
ddct = do_dct(im, n)

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('16x16 DCT bez kwantyzacji oraz transformacja odwrotna')
ax1.imshow(ddct, cmap='gray', vmax = np.max(ddct)*0.01, vmin = 0)
ax2.imshow(undo_dct(ddct, n, imsize) , cmap='gray')
plt.show()

thresh = 0.02
dct_thresh = ddct * (abs(ddct) >= (thresh*np.max(ddct)))
percent_nonzeros = np.sum( dct_thresh != 0.0 ) / dct_thresh.shape[0]**2

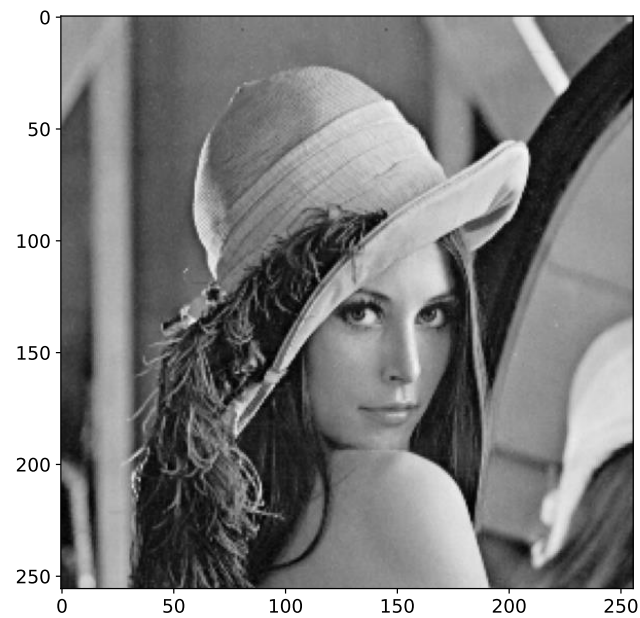
print (f"Keeping only {percent_nonzeros} of the DCT coefficients")

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('16x16 DCT z kwantyzacją oraz transformacja odwrotna')
ax1.imshow(dct_thresh, cmap='gray', vmax = np.max(dct_thresh)*0.01, vmin = 0)
ax2.imshow(undo_dct(dct_thresh, n, imsize) , cmap='gray')
plt.show()

```

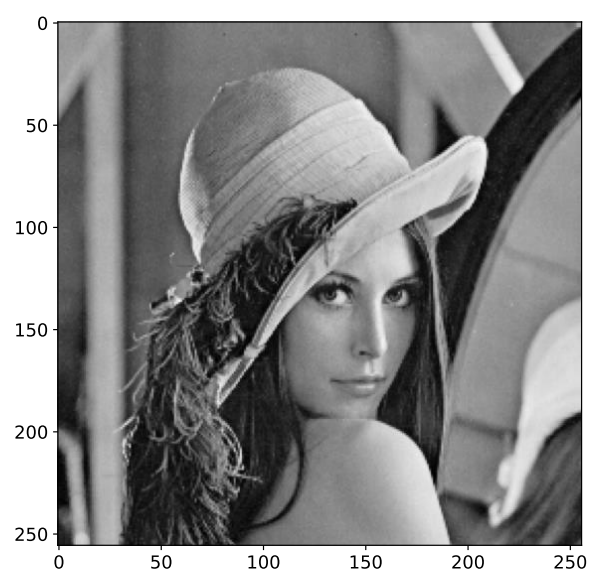
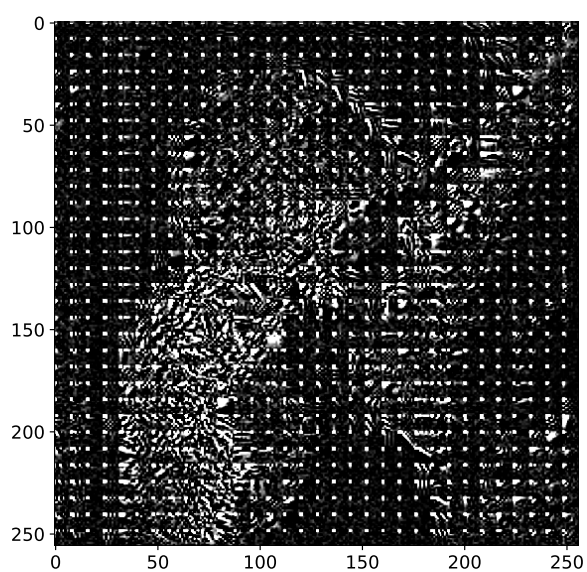
3. Wyniki

Oryginalny obraz *lena.pgm* przedstawia rysunek poniżej.



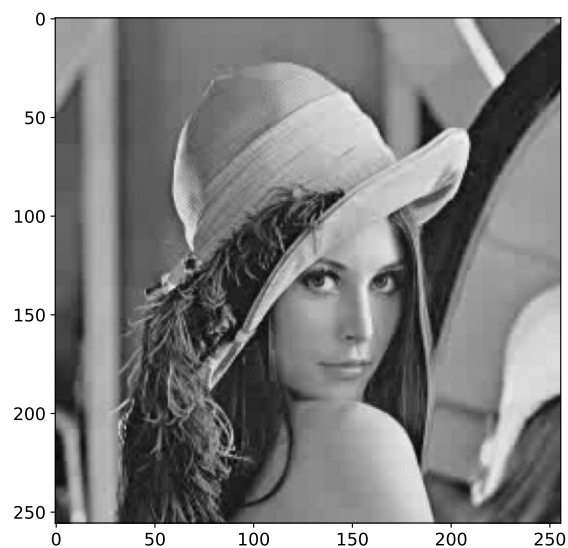
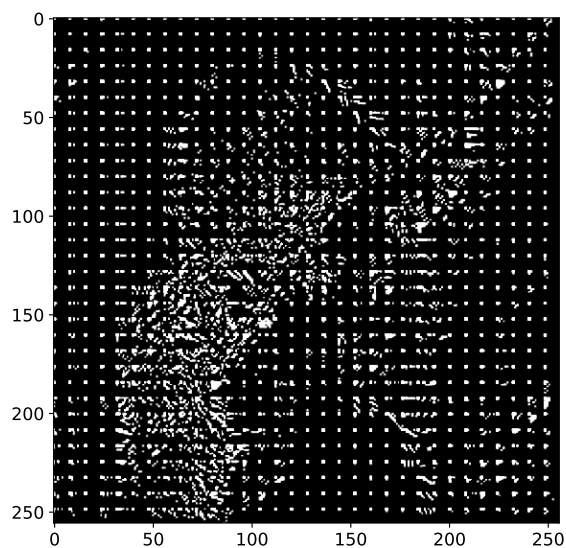
Wynik transformacji DCT bez kwantyzacji w oknie 8x8 oraz jego transformację odwrotną przedstawia rysunek poniżej:

8x8 DCT bez kwantyzacji oraz transformacja odwrotna



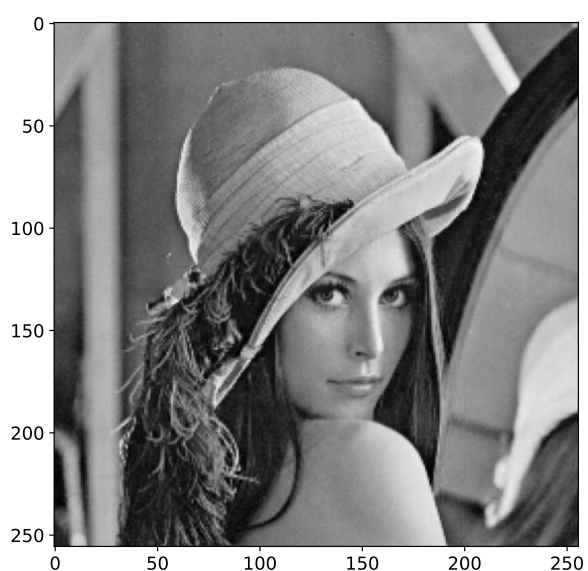
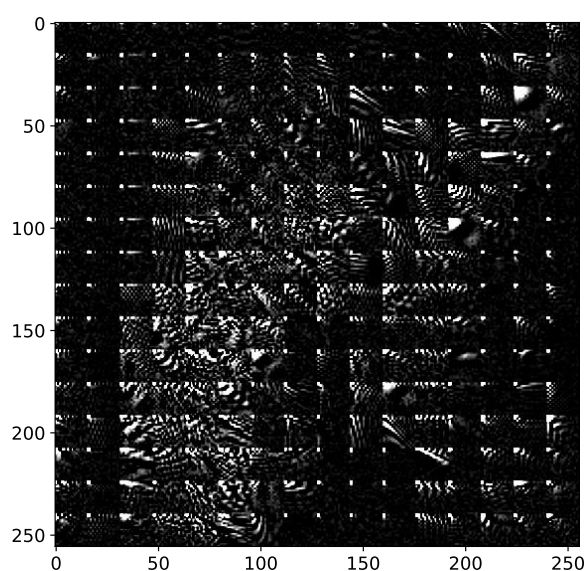
Wynik transformacji DCT w oknie 8x8 z kwantyzacją zostawiającą około 13,3% największych współczynników oraz jego transformację odwrotną przedstawia rysunek poniżej.

8x8 DCT z kwantyzacją oraz transformacja odwrotna



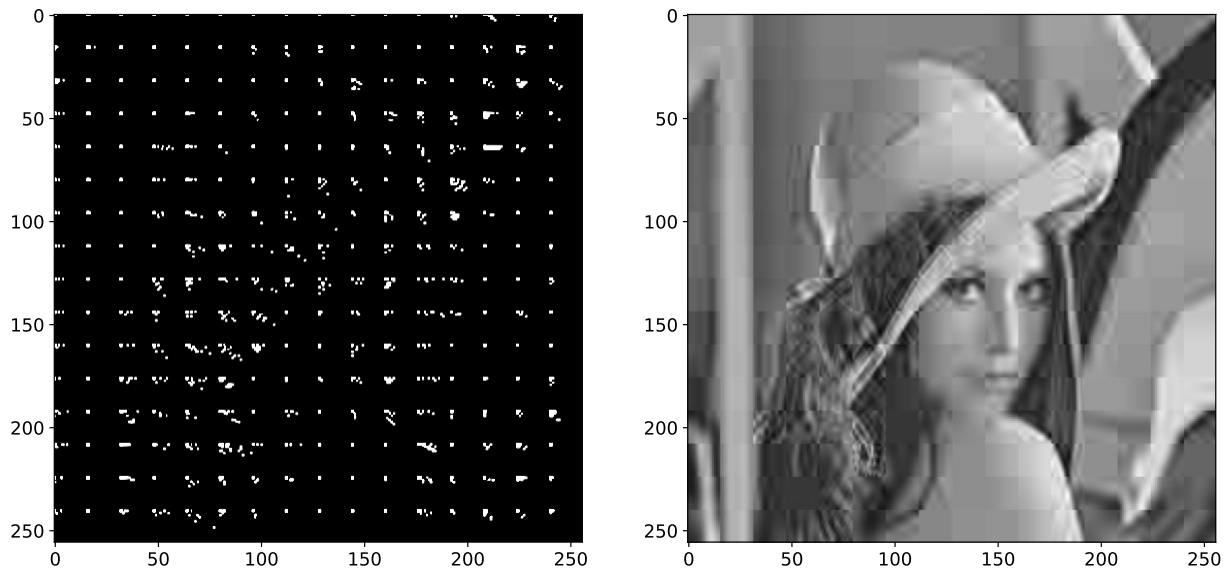
Wynik transformacji DCT bez kwantyzacji w oknie 16x16 oraz jego transformację odwrotną przedstawia rysunek poniżej.

16x16 DCT bez kwantyzacji oraz transformacja odwrotna



Wynik transformacji DCT w oknie 16x16 z kwantyzacją zostawiającą około 2,7% największych współczynników oraz jego transformację odwrotną przedstawia rysunek poniżej

16x16 DCT z kwantyzacją oraz transformacja odwrotna



4. Wnioski

- Dyskretna transformata kosinusowa oraz transformata falkowa mogą być używane do kompresji plików graficznych.
- Rozmiar okna ma wpływ na wynik DCT (im większy tym mniej uwydatnione szczegóły).
- Kwantyzacja pozwala na znaczną kompresję pliku, jednakże wiąże się to z utratą jakości.
- Im bardziej agresywna kwantyzacja tym większa strata na jakości (ale nieliniowa zależność).
- Niewielka ilość największych współczynników transformacji DCT pozwala w pełni odwzorować pierwotny obraz bez znacznej utraty jakości.