

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики.

Лабораторная работа №2
Исследование инвариантности методов сопоставления
изображений в условиях изменчивости
по дисциплине
Компьютерное зрение

Выполнил студент группы М3403:
Давлетов Артем Эдуардович

Преподаватель:
Титаренко Михаил Алексеевич

Цель работы: ознакомиться с методами сопоставления изображений и исследовать их применимость к изображениям, подверженным различным типам изменчивости.

Для определения смещения между изображениями будет использоваться функция `phase_cross_correlation(image, offset_image)`, которая будет принимать на вход два изображения, а будет выдавать три параметра:

`shifts: ndarray`
Вектор сдвига (в пикселях) между двумя изображениями, необходимый для регистрации «offset_image».

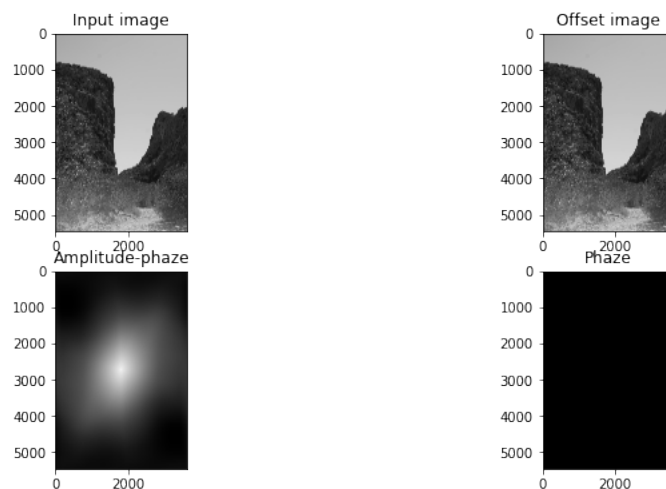
`error: float`
Нормализованная среднеквадратичная ошибка между изображениями «image» and «offset_image».

`phasediff: float`
Глобальная разность фаз между изображениями.

Из всех параметров нам понадобится только `shift`, который будет демонстрировать вектор сдвига между изображениями.

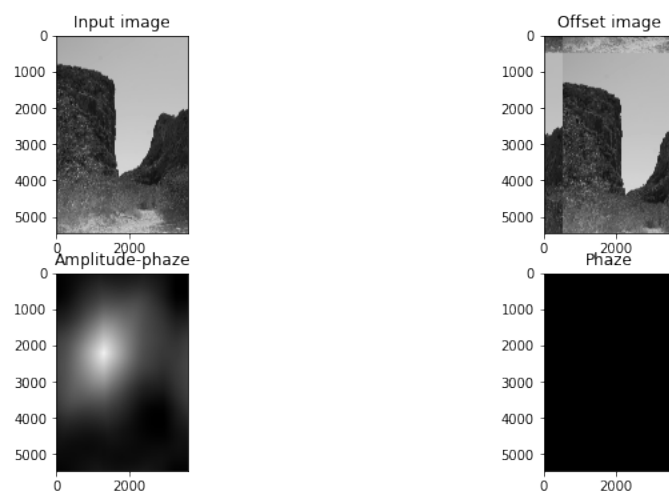
Примеры работы алгоритма амплитудно-фазовой и фазовой корреляции:

1. Методы амплитудно-фазовой и фазовой корреляции для одного и того же изображения.



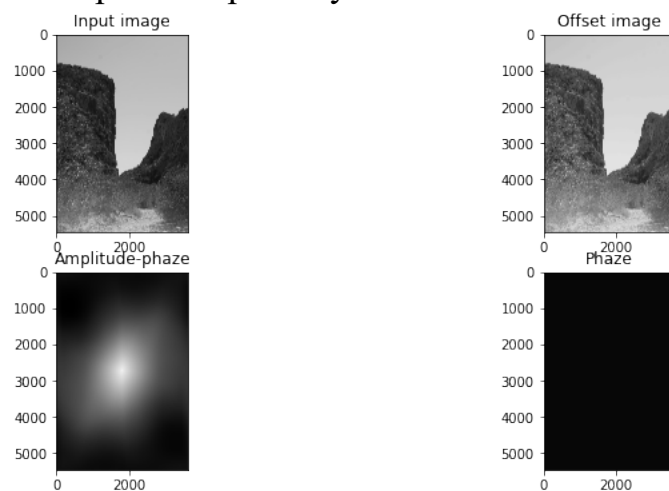
Определенный отступ: [0, 0]

2. Методы амплитудно-фазовой и фазовой корреляции при векторе смещения (500, 500).



Определенный отступ: [500, 500]

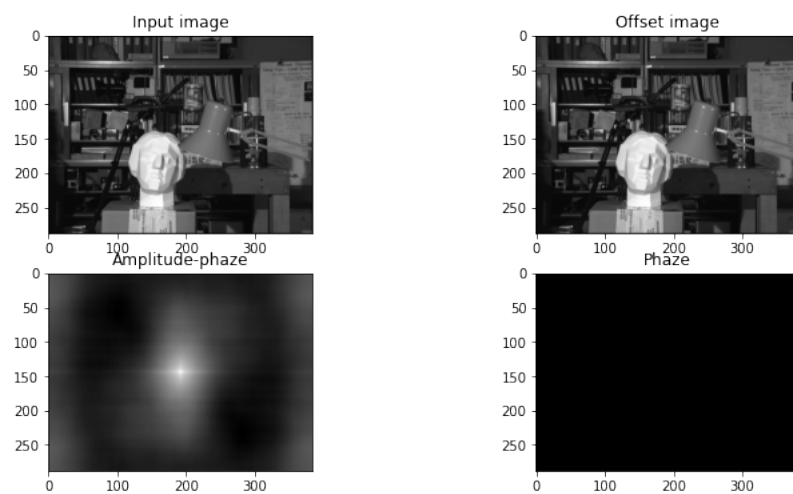
3. Методы амплитудно-фазовой и фазовой корреляции при нулевом векторе смещения в разное время суток.



Определенный отступ: [0, 0]

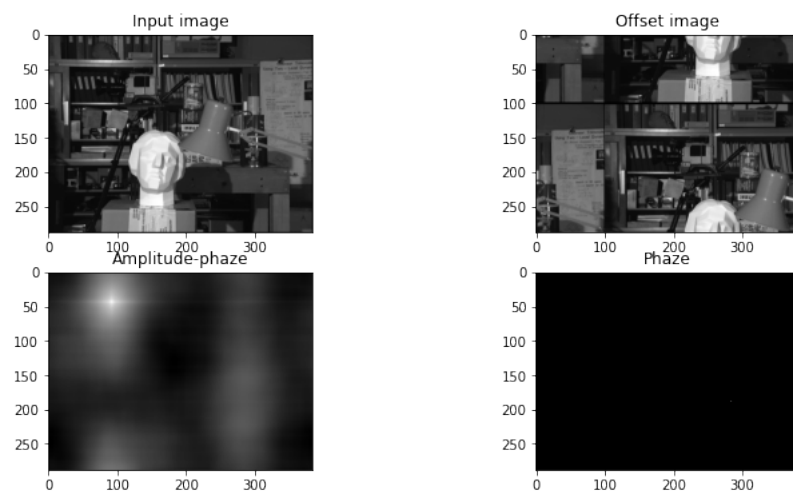
Проверим этот же алгоритм для другого изображения:

1. Методы амплитудно-фазовой и фазовой корреляции для одного и того же изображения.



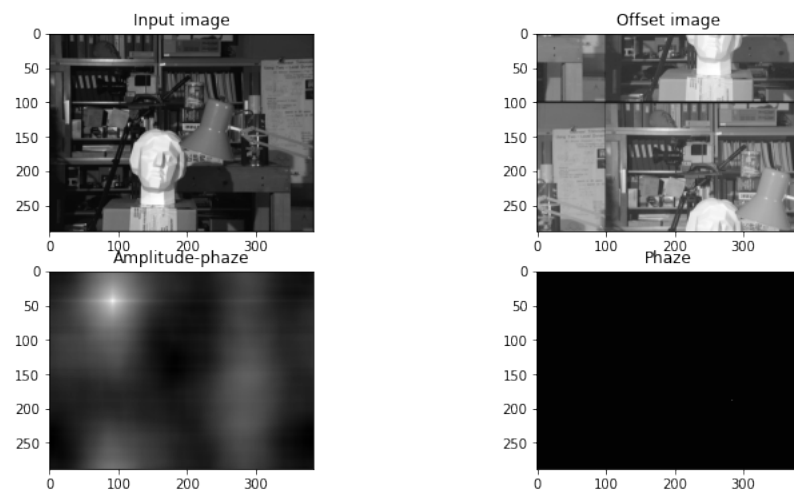
Определенный отступ: [0, 0]

2. Методы амплитудно-фазовой и фазовой корреляции при векторе смещения (100, 100).



Определенный отступ: [100, 100]

3. Методы амплитудно-фазовой и фазовой корреляции при векторе смещения (100, 100) в разное время суток.



Определенный отступ: [100, 100]

Реализация прикреплена ниже в виде jupyter notebook:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

from skimage import data
from skimage import registration
from skimage.registration import phase_cross_correlation
from scipy.ndimage import fourier_shift

def phase_correlation(a, b):
    G_a = np.fft.fft2(a)
    G_b = np.fft.fft2(b)
    conj_b = np.ma.conjugate(G_b)
    R = G_a*conj_b
    R /= np.absolute(R)
    r = np.fft.ifft2(R).real
    return r

def phase_and_show(image, offset_image):
    # pixel precision first
    shift, error, diffphase = phase_cross_correlation(image, offset_image)

    # Amplitude-phase correlation
    image_product = np.fft.fft2(image) * np.fft.fft2(offset_image).conj()
    cc_image = np.fft.fftshift(np.fft.ifft2(image_product))

    # Phaze correlation
    corrimg = phase_correlation(image, offset_image)

    # Show the output
    fig, axes = plt.subplots(2, 2)

    axes[0][0].imshow(image, cmap = 'gray')
    axes[0][0].set_title('Input image')

    axes[0][1].imshow(offset_image.real, cmap = 'gray')
    axes[0][1].set_title('Offset image')

    axes[1][0].imshow(cc_image.real, cmap = 'gray')
    axes[1][0].set_title('Amplitude-phaze')

    axes[1][1].imshow(corrimg, cmap = 'gray')
    axes[1][1].set_title('Phaze')

    fig.set_figwidth(12)
    fig.set_figheight(6)

    plt.show()

    print("Detected pixel offset: {}".format(shift))
```

Вычисление амплитудно-фазовой и фазовой корреляций при нулевом векторе смещения

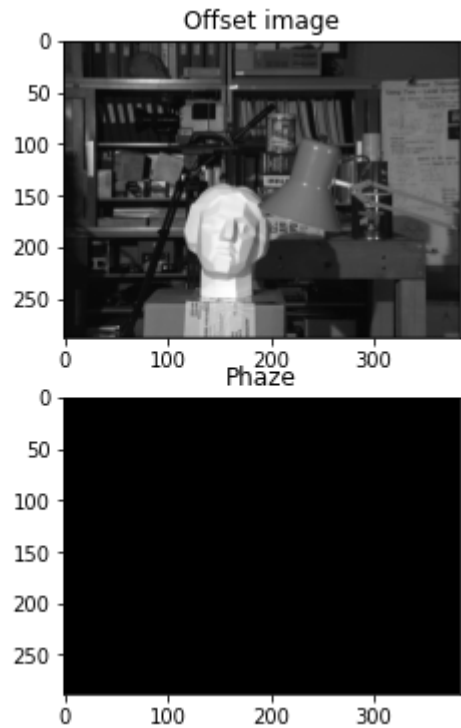
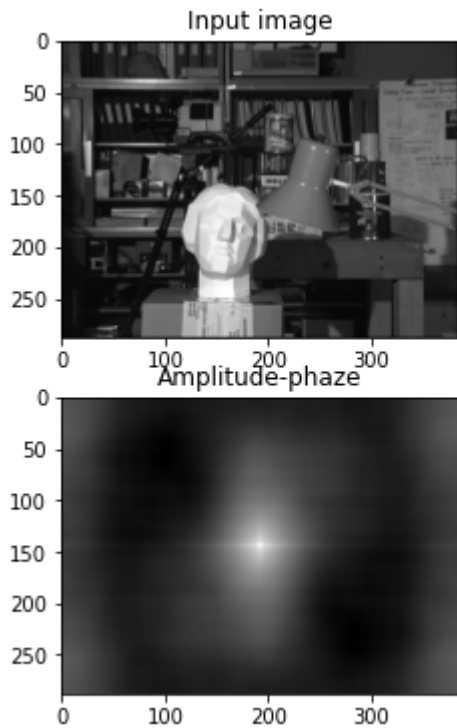
```
image = cv2.imread('left.png',0)
shift = (0, 0)

offset_image = fourier_shift(np.fft.fftn(image), shift)
offset_image = np.fft.ifftn(offset_image)

print("Known offset: {}".format(shift))

phase_and_show(image, offset_image)
```

Known offset: (0, 0)



Detected pixel offset: [0. 0.]

Вычисление амплитудно-фазовой и фазовой корреляций при векторе смещения (100, 100)

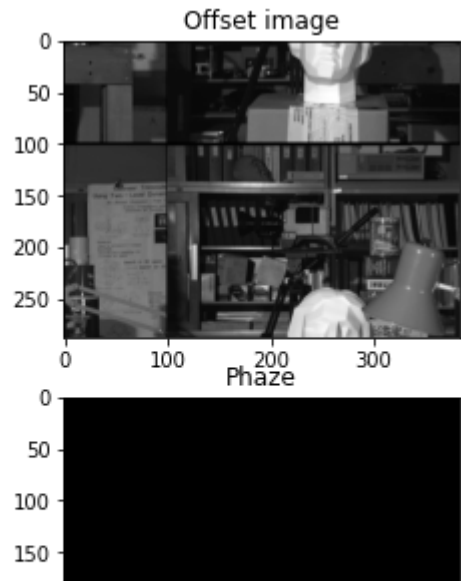
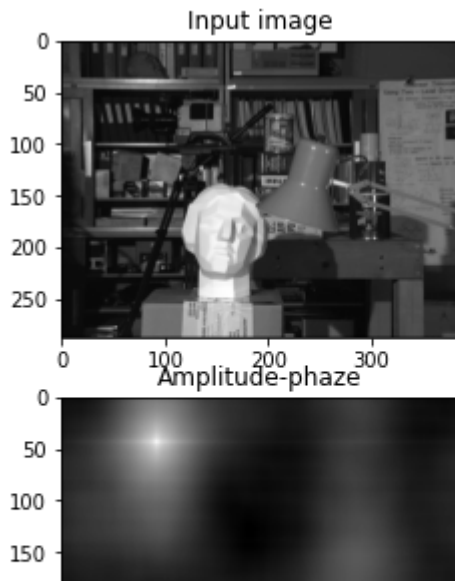
```
image = cv2.imread('left.png',0)
shift = (100, 100)

offset_image = fourier_shift(np.fft.fftn(image), shift)
offset_image = np.fft.ifftn(offset_image)

print("Known offset: {}".format(shift))

phase_and_show(image, offset_image)
```

Known offset: (100, 100)



Вычисление амплитудно-фазовой и фазовой корреляций при векторе смещения (100, 100) в разное время суток

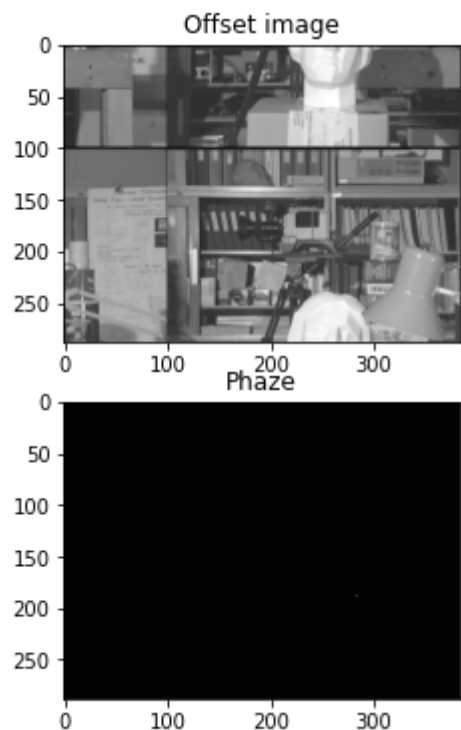
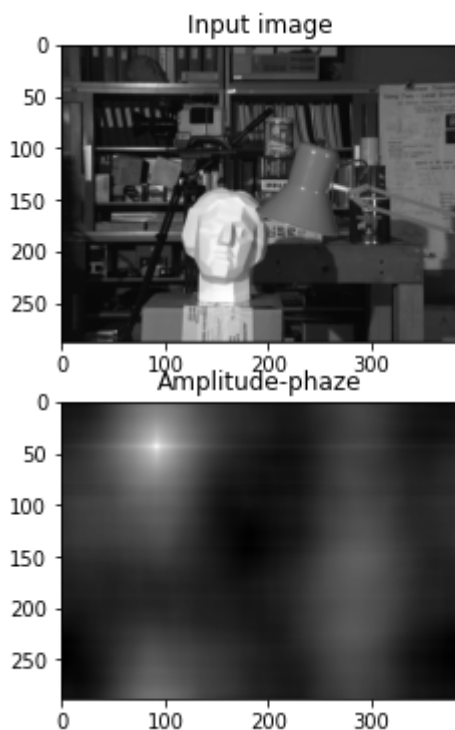
```
image = cv2.imread('left.png',0)
shift = (100, 100)

offset_image = (255.0/3)*(image/(255.0/2))*0.5
offset_image = fourier_shift(np.fft.fftn(offset_image), shift)
offset_image = np.fft.ifftn(offset_image)

print("Known offset: {}".format(shift))

phase_and_show(image, offset_image)
```

Known offset: (100, 100)



Detected pixel offset: [-100. -100.]

Вычисление амплитудно-фазовой и фазовой корреляций при нулевом векторе смещения

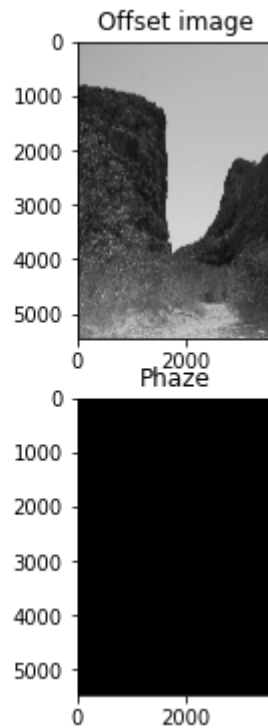
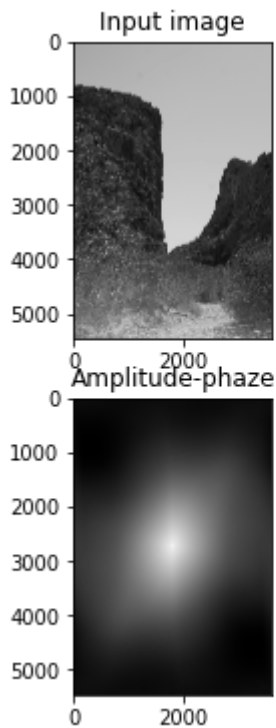
```
image = cv2.imread('img.JPG',0)
shift = (0, 0)

offset_image = fourier_shift(np.fft.fftn(image), shift)
offset_image = np.fft.ifftn(offset_image)

print("Known offset: {}".format(shift))

phase_and_show(image, offset_image)
```

Known offset: (0, 0)



Detected pixel offset: [0. 0.]

Вычисление амплитудно-фазовой и фазовой корреляций при векторе смещения (500, 500)

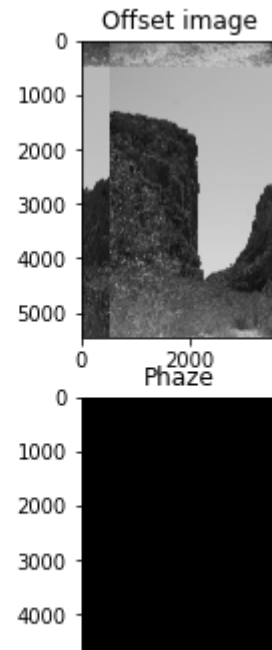
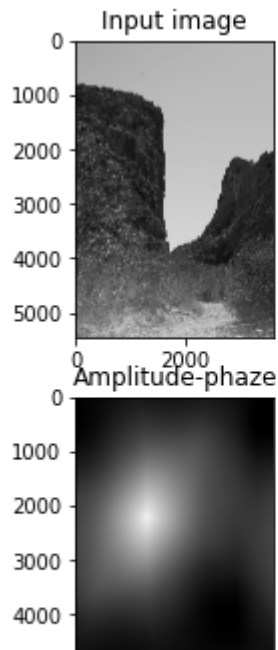
```
image = cv2.imread('img.JPG',0)
shift = (500, 500)

offset_image = fourier_shift(np.fft.fftn(image), shift)
offset_image = np.fft.ifftn(offset_image)

print("Known offset: {}".format(shift))

phase_and_show(image, offset_image)
```

Known offset: (500, 500)

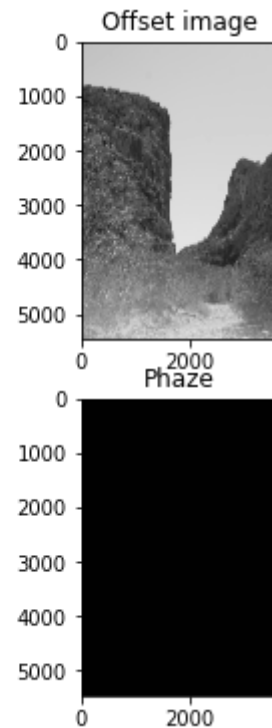
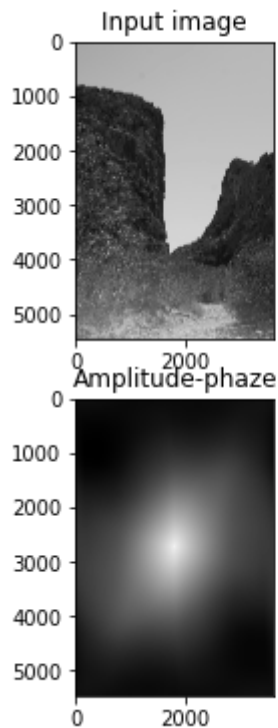


Вычисление амплитудно-фазовой и фазовой корреляций при нулевом векторе смещения в разное время суток

```
image = cv2.imread('img.JPG',0)
shift = (0, 0)

offset_image = (255.0/3)*(image/(255.0/2))*0.5

phase_and_show(image, offset_image)
```



Detected pixel offset: [0. 0.]