**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная работа №1

**Случайные процессы в обработке изображений**

Вариант 2

**Выполнили:**

Крученков Евгений Андреевич,

Кендысь Алексей Максимович,

студенты 4 курса, 7 группы,

специальность

“прикладная математика”

**Преподаватель:**

Старший преподаватель

кафедры ТВиМС ФПМИ,

Л.А. Хаткевич

Минск, 2023

**СОДЕРЖАНИЕ**

Задание 1. Чтение изображения 2

Задание 2. Выделение красного канала, поворот изображения 2

Задание 3. Модуль разности изображений 5

Вывод 6

Задание 1. Чтение изображения

Подключение библиотек и задание параметров.

import math

import os

import array as arr

import numpy as np

import pandas as pd

import scipy as sp

import matplotlib.pyplot as plt

degr = 37

Чтение изображения из файла.

image = plt.imread("Bobyor.jpg").copy()

Вывод изображения.

def show\_image(ax, image, title=None, cmap=None):

    ax.imshow(image, cmap=cmap)

    ax.set\_title(title)

    ax.xaxis.set\_visible(False)

    ax.yaxis.set\_visible(False)

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image, title="Бобр", cmap="gray")



Задание 2. Выделение красного канала, поворот изображения

Выделение красного канала.

size = image.shape

image\_red = [0]\*size[0]

for i in range(size[0]):

    image\_red[i] = [0]\*size[1]

for i in range(size[0]):

    for j in range(size[1]):

        image\_red[i][j] = image[i][j][0]

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_red, title="Бобр (красный)", cmap="gray")



Поворот красного канала на 37 градусов по часовой стрелке.

def rotate\_image(image, degree, is\_clockwise):

    rads = math.radians(degree)

    rot\_img = [0]\*size[0]

    for i in range(size[0]):

        rot\_img[i] = [0]\*size[1]

    midx, midy = (size[0]//2, size[1]//2)

    for i in range(size[0]):

        for j in range(size[1]):

            if (is\_clockwise):

                x = (i-midx)\*math.cos(rads)-(j-midy)\*math.sin(rads)

                y = (i-midx)\*math.sin(rads)+(j-midy)\*math.cos(rads)

            else:

                x = (i-midx)\*math.cos(rads)+(j-midy)\*math.sin(rads)

                y = -(i-midx)\*math.sin(rads)+(j-midy)\*math.cos(rads)

            x = round(x)+midx

            y = round(y)+midy

            if (x>=0 and y>=0 and x<size[0] and  y<size[1]):

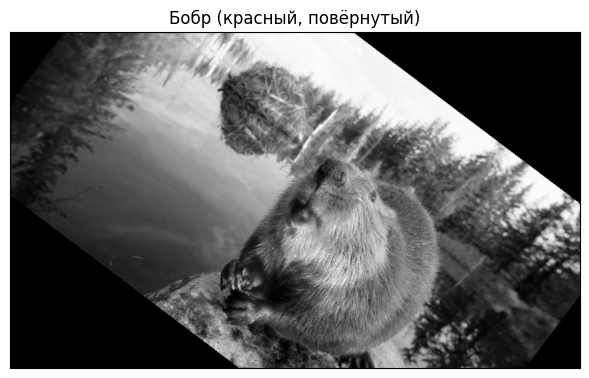
                rot\_img[i][j] = image[x][y]

    return rot\_img

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

rotated\_image\_red = rotate\_image(image\_red, degr, True)

show\_image(ax, rotated\_image\_red, title="Бобр (красный, повёрнутый)", cmap="gray")



Поворот изображения на 37 градусов против часовой стрелки.

rotated\_again\_red = rotate\_image(rotated\_image\_red, degr, False)

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, rotated\_again\_red, title="Бобр (красный, повёрнутый обратно)", cmap="gray")



Задание 3. Модуль разности изображений

Модуль разности исходного и полутонового изображения.

image\_red\_diff = [0]\*size[0]

for i in range(size[0]):

    image\_red\_diff[i] = [0]\*size[1]

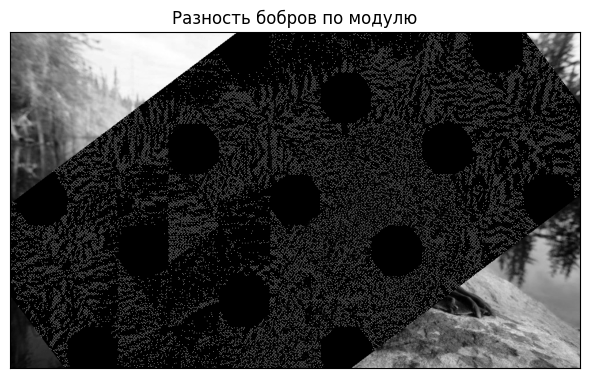
for i in range(size[0]):

    for j in range(size[1]):

        image\_red\_diff[i][j] = abs(image\_red[i][j]-rotated\_again\_red[i][j])

fig, ax = plt.subplots(figsize=(6, 6), layout="tight")

show\_image(ax, image\_red\_diff, title="Разность бобров по модулю", cmap="gray")



Вывод

При повороте происходит обрезка изображения, т.к. высота и ширина изображения не меняется. Соответственно, при повороте углы обрезаются, а углы изображения модуля разности совпадают с исходным красным каналом.

Что касается остальной части изображения, мы можем наблюдать погрешность при вычислении поворота. Т.к. при повороте используется округление для правильного задания координат, то некоторые значения матрицы изображения при повороте назад будут не полностью совпадать с исходными. Следовательно, в изображении модуля разности видны черные/темно-серые цвета, соответствующие погрешности округления.