Тестовое задание для ГосНИИАС
Выполнил: Карасик Андрей

1 Создание класса генерации прямоугольников

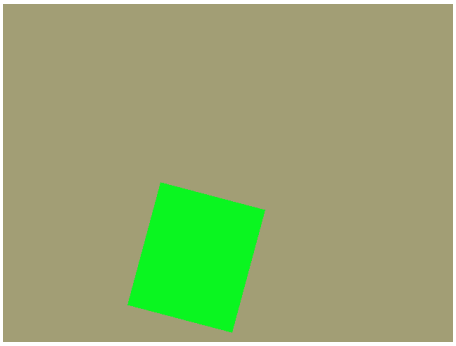
Листинг кода[1]:

```
from PIL import Image, ImageDraw
import random
import numpy as np


class Rectangle:
    """
        Rectangle generation class:
        create_rec: gives coordinates of rectangle,
                    height, width and coordinates of top-left angle of surrounding rectangle
        image_gen: generates, saves and show our rectangle
    """
    def __init__(self, gen, bound):
        self.coords_gen = gen
        self.coords_bound = bound
        self.save = None
        self.background = None
        self.create_rect()
        self.image_gen()

    def create_rect(self):
        w = np.random.randint(150, 250)  # rectangle width
        h = np.random.randint(150, 250)  # rectangle height
        angle = random.randint(0, 89) * 3.14159 / 180  # rotation angle

        h1 = (h * np.cos(angle))
        h2 = (w * np.sin(angle))
        hhh = h1 + h2  # surrounding rectangle height
        w1 = (h * np.sin(angle))
        w2 = (w * np.cos(angle))
        www = w1 + w2  # surrounding rectangle width

        x0, y0 = (np.random.randint(www/2, 640-www/2), np.random.randint(hhh/2, 480-hhh/2))  # center

        # surrounding angle coordinates
        xb1yb1, xb2yb2, xb3yb3, xb4yb4 = ((x0 - www/2, y0 - hhh/2),
                                          (x0 - www/2, y0 + hhh/2),
                                          (x0 + www/2, y0 + hhh/2),
                                          (x0 + www/2, y0 - hhh/2))

        rotation_matrix = np.array([[np.cos(angle), -np.sin(angle)],
                                    [np.sin(angle), np.cos(angle)]])
        # angle coordinates
        x01y01 = (x0 + (x0 - w / 2 - x0) * rotation_matrix[0][0] + (y0 - h / 2 - y0) * rotation_matrix[0][1],
                  y0 + (x0 - w / 2 - x0) * rotation_matrix[1][0] + (y0 - h / 2 - y0) * rotation_matrix[1][1])
        x02y02 = (x0 + (x0 + w / 2 - x0) * rotation_matrix[0][0] + (y0 - h / 2 - y0) * rotation_matrix[0][1],
                  y0 + (x0 + w / 2 - x0) * rotation_matrix[1][0] + (y0 - h / 2 - y0) * rotation_matrix[1][1])
        x03y03 = (x0 + (x0 + w / 2 - x0) * rotation_matrix[0][0] + (y0 + h / 2 - y0) * rotation_matrix[0][1],
                  y0 + (x0 + w / 2 - x0) * rotation_matrix[1][0] + (y0 + h / 2 - y0) * rotation_matrix[1][1])
        x04y04 = (x0 + (x0 - w / 2 - x0) * rotation_matrix[0][0] + (y0 + h / 2 - y0) * rotation_matrix[0][1],
                  y0 + (x0 - w / 2 - x0) * rotation_matrix[1][0] + (y0 + h / 2 - y0) * rotation_matrix[1][1])

        self.coords_gen = (x01y01, x02y02, x03y03, x04y04)
        self.coords_bound = xb2yb2, www, hhh

    def image_gen(self):
        # background with random color
        background = Image.new('RGB', (640, 480), (random.randint(0, 255),
```

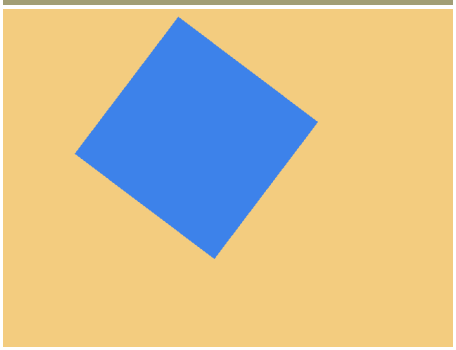
[1]весь код размещен с комментариями *в данном репозитории*

```
                                                random.randint(0, 255),
                                                random.randint(0, 255)))
        rectangle = ImageDraw.Draw(background)
        # rectangle with random color
        rectangle.polygon(self.coords_gen, (random.randint(0, 255),
                                            random.randint(0, 255),
                                            random.randint(0, 255)))
        background.show()
        name = '/home/andrei/PycharmProjects/test_task_GosNIIAS/rect_pics/6.png'
        background.save(name, 'png')


if __name__ == "__main__":
    data = open('data.txt', 'w')  # shape data file
    for i in range(0, 1):
        rect = Rectangle(None, None)
        print(f"Bounding rectangle:\n"
              f"x = {rect.coords_bound[0][0]}\n"
              f"y = {rect.coords_bound[0][1]}\n"
              f"w = {rect.coords_bound[1]}\n"
              f"h = {rect.coords_bound[2]}\n\n"
              f"Angles coordinates:\n"
              f"x1 = {rect.coords_gen[0][0]}, y1 = {rect.coords_gen[0][1]}\n"
              f"x2 = {rect.coords_gen[1][0]}, y2 = {rect.coords_gen[1][1]}\n"
              f"x3 = {rect.coords_gen[2][0]}, y3 = {rect.coords_gen[2][1]}\n"
              f"x4 = {rect.coords_gen[3][0]}, y4 = {rect.coords_gen[3][1]}\n"
              f"**********\n", file=data)
    data.close()
```

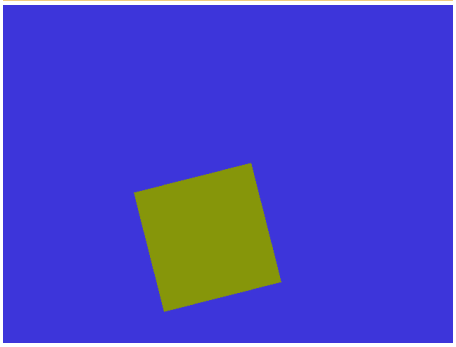
Примеры:



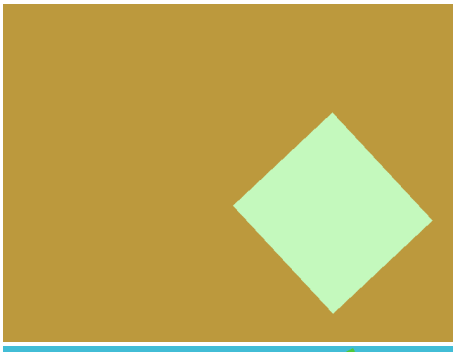
Bounding rectangle:
x = 177.81297507509635 y = 466.7329701271386
w = 194.3740498498073 h = 213.46594025427723
Angles coordinates:
x1 = 224.40036474598526, y1 = 253.2670298728614
x2 = 372.18702492490365, y2 = 292.8663110931169
x3 = 325.59963525401474, y3 = 466.7329701271386
x4 = 177.81297507509635, y4 = 427.1336889068831



Bounding rectangle:
x = 102.84868366448603 y = 354.6592632085511
w = 344.30263267102794 h = 343.31852641710225
Angles coordinates:
x1 = 249.08962843382267, y1 = 11.34073679144889
x2 = 447.15131633551397, y2 = 160.59075449842615
x3 = 300.9103715661774, y3 = 354.6592632085511
x4 = 102.84868366448605, y4 = 205.40924550157382



Bounding rectangle:
x = 186.99243908133724 y = 435.1001196274153
w = 208.01512183732552 h = 210.20023925483062
Angles coordinates:
x1 = 352.91296192481013, y1 = 224.8998803725847
x2 = 395.00756091866276, y2 = 393.73128958173254
x3 = 229.08703807518987, y3 = 435.1001196274153
x4 = 186.99243908133724, y4 = 266.26871041826746



Bounding rectangle:
x = 327.155531813269 y = 439.2393030848971
w = 283.6889363734619 h = 284.4786061697942
Angles coordinates:
x1 = 468.3067050246973, y1 = 154.76069691510287
x2 = 610.8444681867309, y2 = 307.61352179172724
x3 = 469.6932949753027, y3 = 439.23930308489713
x4 = 327.155531813269, y4 = 286.38647820827276

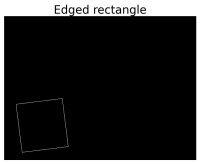
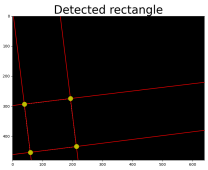
Bounding rectangle:
x = 348.54686848903714 y = 236.48681849581902
w = 230.90626302192572 h = 232.97363699163805
Angles coordinates:
x1 = 497.76493916360744, y1 = 3.51318150418097
x2 = 579.4531315109629, y2 = 157.1459879975256
x3 = 430.23506083639256, y3 = 236.48681849581902
x4 = 348.54686848903714, y4 = 82.8540120024744

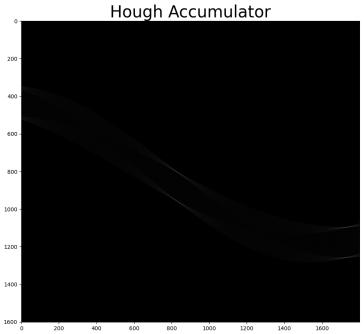
2 Поиск координат углов прямоугольника при помощи преобразования Хафа

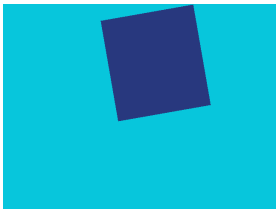
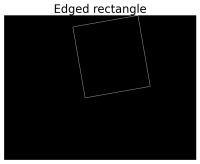
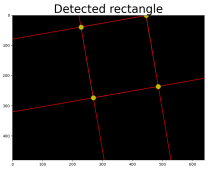
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.transform import hough_line, hough_line_peaks


def find_inter(k1, b1, k2, b2):
    """
        y = k1 * x + b1
        y = k2 * x + b2
    """
    if k2 != k1:
        x = - (b2 - b1) / (k2 - k1)
        y = k1 * x + b1
        return x, y
    else:
        return None


image = cv2.imread('/home/andrei/PycharmProjects/test_task_GosNIIAS/rect_pics/6.png')  # open image
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)  # to gray
gray = cv2.GaussianBlur(gray, (3, 3), 0)  # some blur
cv2.imwrite("gray.png", gray)  # save gray image

edged = cv2.Canny(gray, 0, 10)  # leave only the borders
cv2.imwrite("edged.png", edged)   # save edged image

using_angles = np.linspace(-np.pi / 2, np.pi / 2, 1800)  # introduce angles array

halfspace, theta, dist = hough_line(edged, using_angles)

# draw Hough Accumulator
plt.figure(figsize=(15, 10))
plt.title('Hough Accumulator', size=30)
plt.imshow(halfspace, cmap='gray')
plt.savefig('accumulator.png')

angle_list = []

fig, axes = plt.subplots(1, 2, figsize=(20, 8))
ax = axes.ravel()

# draw edged rectangle
ax[0].imshow(edged, cmap='gray')
ax[0].set_title('Edged rectangle', size=30)
ax[0].set_axis_off()

ax[1].imshow(image, cmap='gray')

origin = np.array((0, edged.shape[1]))
lines = []

# draw straight lines along the borders using Hough Transform
for _, angle, dist in zip(*hough_line_peaks(halfspace, theta, dist)):
    y0, y1 = (dist - origin * np.cos(angle)) / np.sin(angle)
    ax[1].plot(origin, (y0, y1), '-r')
    lines += [(1 / np.tan(angle), dist / np.sin(angle))]
    print(np.degrees(angle), dist)

inter = []

# lines intersection
for i in range(len(lines)):
    for j in range(i+1, len(lines)):
        print(f"intersection of lines{lines[i]} and {lines[j]}:", end=' ')
        point = find_inter(lines[i][0], lines[i][1], lines[j][0], lines[j][1])
        if point != None:
```

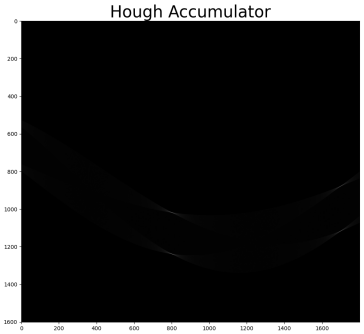
```
            x, y = point
            if abs(x) > 1000 or abs(y) > 1000:
                print("NO INTERSECTION")
            else:
                print(x, y)
                inter += [(-x, y)]

# draw intersection points
ax[1].plot([i[0] for i in inter], [i[1] for i in inter], 'yo', markersize=12)
ax[1].set_xlim(origin)
ax[1].set_ylim((edged.shape[0], 0))
ax[1].set_title('Detected rectangle', size=30)
plt.savefig('detection.png')
plt.show()
```

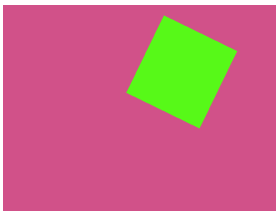
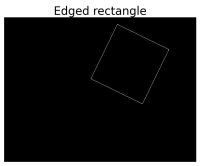
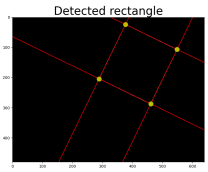
Примеры:

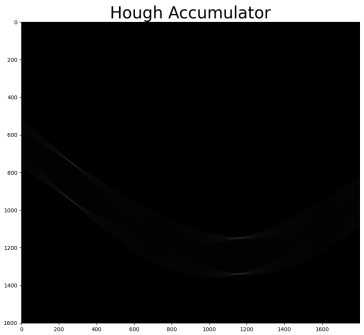
1. 

Edged rectangle

Detected rectangle


Hough Accumulator

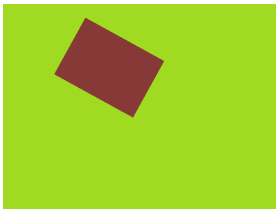
2. 

Edged rectangle

Detected rectangle

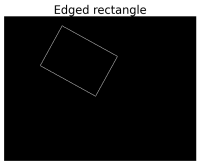

Hough Accumulator

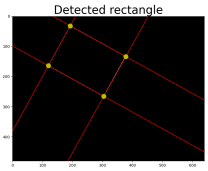
3. 

Edged rectangle

Detected rectangle

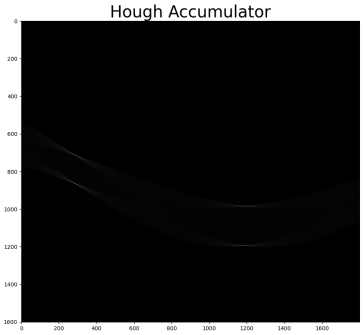

Hough Accumulator

4.

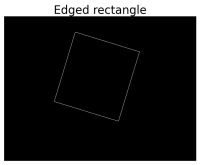


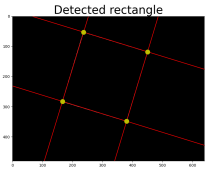

Edged rectangle

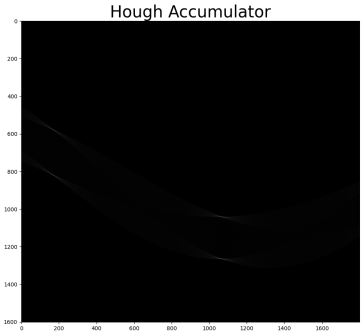

Detected rectangle


Hough Accumulator

5.


Edged rectangle


Detected rectangle


Hough Accumulator