

Федеральное государственное автономное образовательное учреждение высшего  
образования «Национальный исследовательский университет ИТМО»

## **Лабораторная работа №1**

По дисциплине «Алгоритмы компьютерной графики»

**Малышев Никита Александрович**

Академическая группа: Р3312

ИСУ: 409067

Санкт-Петербург  
2025 год

## **Содержание**

1. Выполнение .....	3
1.1. Код .....	3
1.2. Работа программы .....	5
1.2.1. Начальное состояние .....	5
1.2.2. Изображение №1 .....	5
1.2.2.1. Изображение .....	5
1.2.2.2. Среднее значение RGB .....	6
1.2.3. Изображение №2 .....	6
1.2.3.1. Изображение .....	6
1.2.3.2. Среднее значение RGB .....	7
2. Вывод .....	7

# 1. Выполнение

## 1.1. Код

```
import tkinter as tk
from PIL import Image, ImageTk
import matplotlib.pyplot as plt

class App:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Image Switcher + RGB Average")
        self.root.geometry("800x600")

        self.frame = tk.Frame(self.root)
        self.frame.pack(pady=10)

        self.but1 = tk.Button(self.frame, text="image 1", command=lambda:
self.load_image("img/img1.jpeg"))
        self.but1.pack(side="left", padx=10)

        self.but2 = tk.Button(self.frame, text="image 2", command=lambda:
self.load_image("img/img2.jpeg"))
        self.but2.pack(side="left", padx=10)

        self.canvas = tk.Canvas(self.root, bg="white")
        self.canvas.pack(fill="both", expand=True)

        self.original_image = None
        self.photo = None

        self.current_figure = None

        self.root.bind("<Configure>", self.on_resize)

        self.root.mainloop()

    def load_image(self, path):
        self.original_image = Image.open(path).convert("RGB")
        self.update_image()
        self.show_rgb_chart()

    def update_image(self):
        if self.original_image is None:
            return

        w = self.canvas.winfo_width()
        h = self.canvas.winfo_height()

        if w > 1 and h > 1:
            img = self.original_image.copy()
            img.thumbnail((w, h), Image.LANCZOS)
            self.photo = ImageTk.PhotoImage(img)

            self.canvas.delete("all")
            self.canvas.create_image(w // 2, h // 2, anchor="center",
image=self.photo)
```

```

def count_rgb_average(self):
    if self.original_image is None:
        return 0, 0, 0

    pixels = self.original_image.getdata()
    r = g = b = 0
    total = len(pixels)

    for pr, pg, pb in pixels:
        r += pr
        g += pg
        b += pb

    return r / total, g / total, b / total

def show_rgb_chart(self):
    if self.current_figure is not None:
        plt.close(self.current_figure)

    r, g, b = self.count_rgb_average()
    self.current_figure = plt.figure("RGB averages")
    plt.bar(["R", "G", "B"], [r, g, b], color=["red", "green", "blue"])
    plt.title("Average RGB values")
    plt.ylim(0, 255)
    plt.show(block=False)

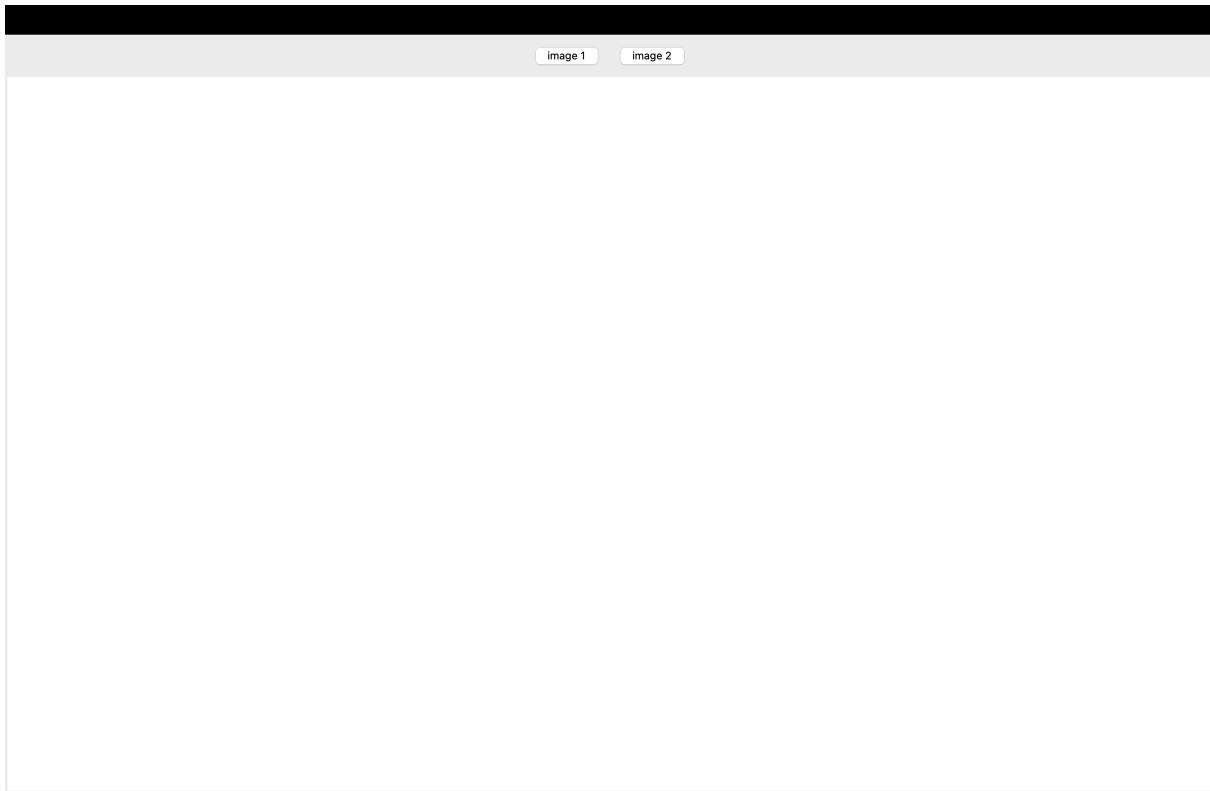
def on_resize(self, event):
    self.update_image()

app = App()

```

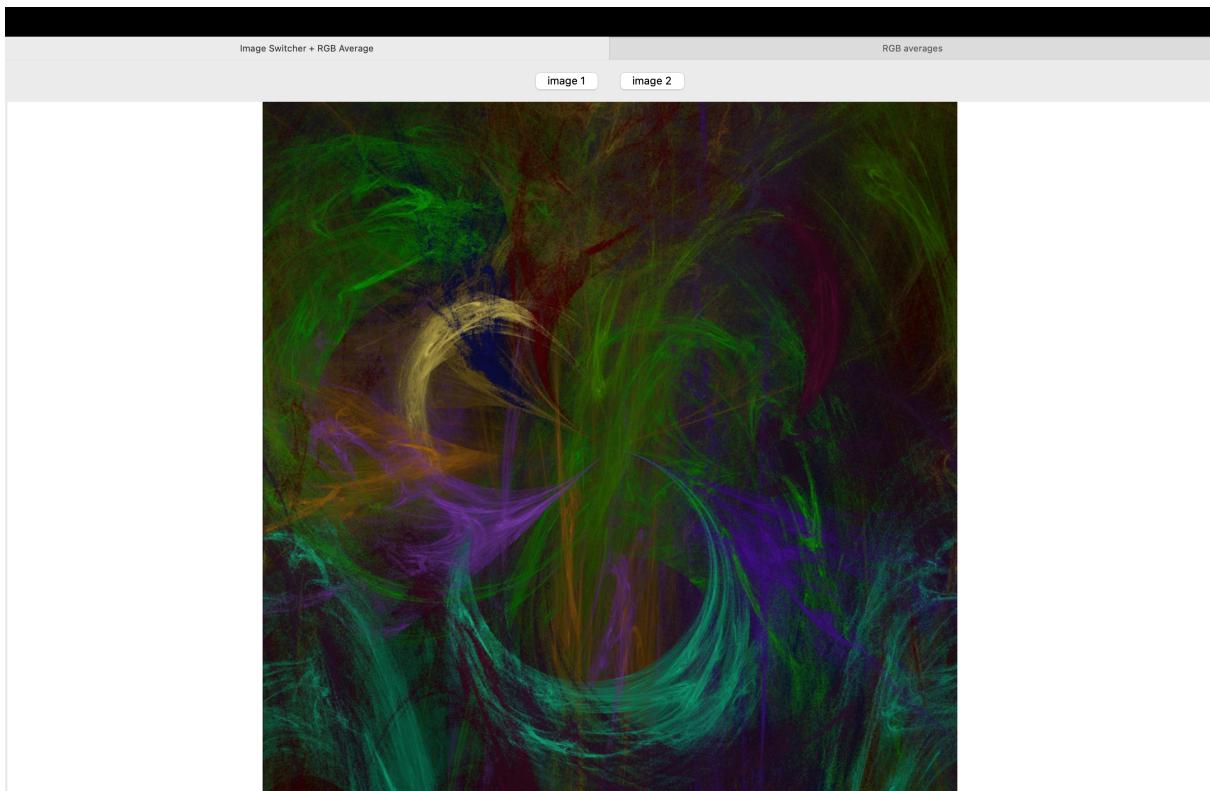
## **1.2. Работа программы**

### **1.2.1. Начальное состояние**

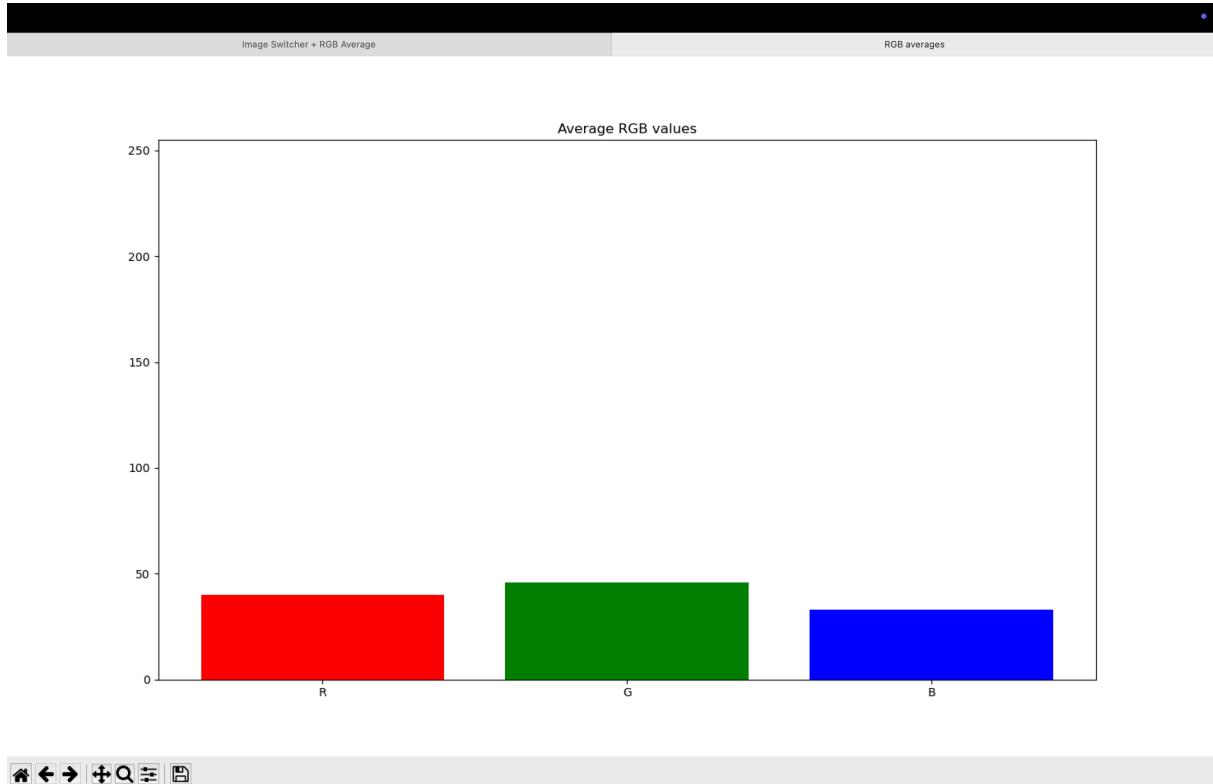


### **1.2.2. Изображение №1**

#### **1.2.2.1. Изображение**

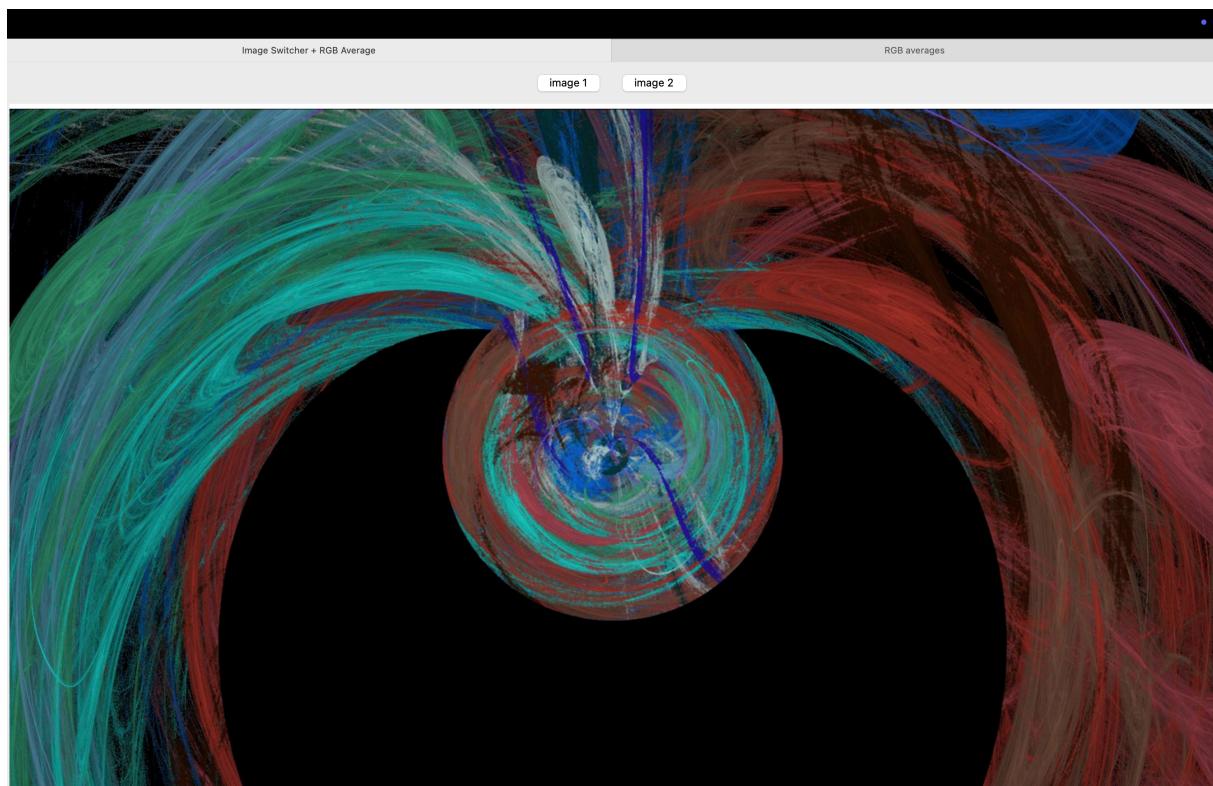


### 1.2.2.2. Среднее значение RGB

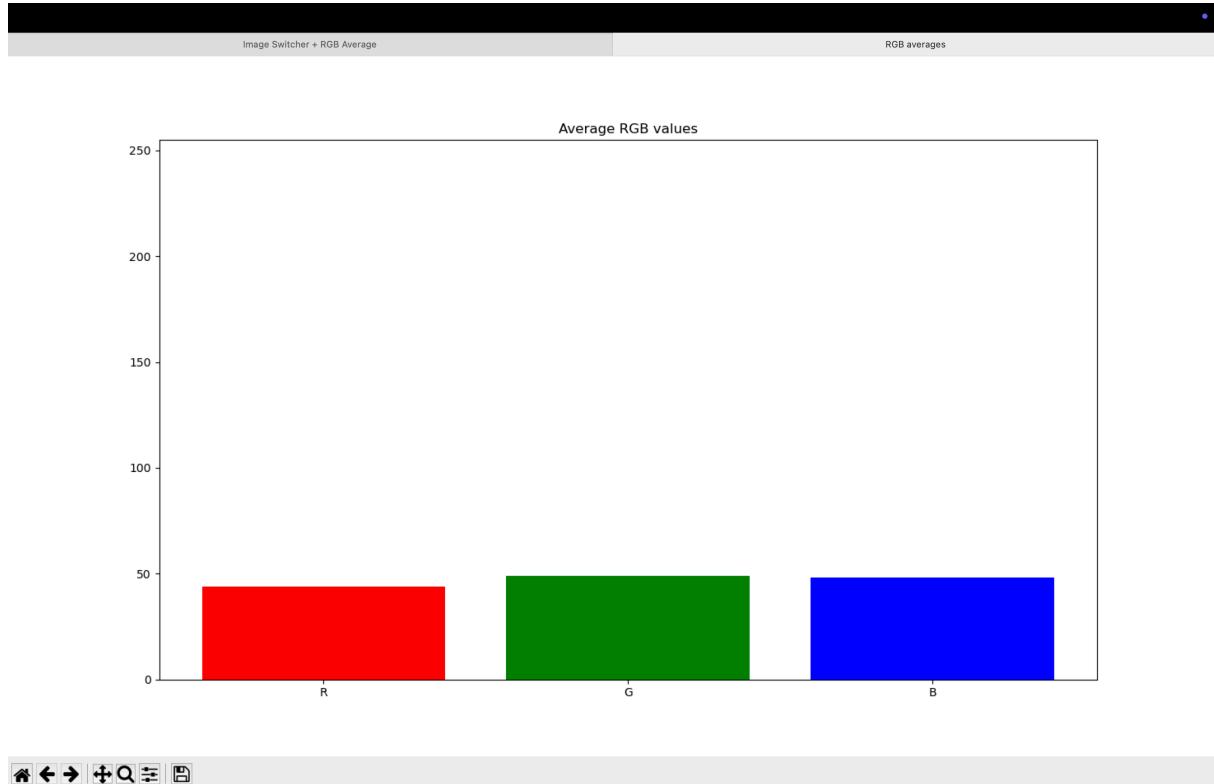


### 1.2.3. Изображение №2

#### 1.2.3.1. Изображение



### 1.2.3.2. Среднее значение RGB



## 2. Вывод

Данная лабораторная работа помогла мне познакомиться с важным понятием среднего значения RGB. А также я научился основам работы с библиотеками `tkinter` и `PIL` для работы с изображениями в Python.