Санкт-Петербургский государственный электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МО ЭВМ

Отчет

по лабораторной работе №1

**«Геометрические преобразования изображений»**

Выполнил: Осмоловский П.Д.

Группа: 4303

Факультет: КТИ

Преподаватель: Черниченко Д.А.

*Санкт-Петербург*

*2017*

**Задача**

Реализовать на языке Python с использованием библиотеки OpenCV программу, выполняющее геометрические преобразования изображения.

**Входные данные**

1. Цветное изображение в формате bmp, jpg
2. Угол поворота - целое число
3. Коэффициент - рациональное число
4. Признак используемой интерполяционной схемы ( 0 – нулевого порядка, 1 – первого порядка, 2- третьего порядка).

**Выходные данные**

Цветное изображение в формате bmp (result.bmp), являющиеся результатом следующих пребразований:

1. Исходное избражение поворачивается на  градусов против часовой стрелке
2. Из получившегося изображения вырезается прямоугольник максимальной площади, вписаный в повернутое изображение
3. Данное изображение увеличивается по оси Y в раз и уменьшается по оси Х в раз.
4. Результат с использванием интерполяции соответствующего порядка записывается в файл result.bmp

**Ход работы**

Для решения поставленных задач я установил пакет OpenCV, источник в списке литературы. Далее добавил в проект два тестовых изображения разного размера: big.jpg и tram.jpg.

При выполнении основной части работы было выполнено:

1. Размеры изображения были пропорционально изменены для комфортной работы (w = 500)
2. Изображение было повернуто на заданный угол
3. Для каждой точки на границе рисунка была найдена ее новая позиция. Результат представлен в виде матрицы, где точки, принадлежащие границе рисунка, имеют значение “1”, а остальные “-1000”.
4. Были отсечены значения, выходящие за рамки экранной формы, эти значения перенесены на границы экранной формы, получена матрица, где “1” выделен многоугольник
5. Многоугольник заполняется “1”, получаем матрицу -1000 и 1, где 1 – точка принадлежащая повернутому изображению
6. Для полученной матрицы воспользуемся алгоритмом Кадане для поиска подматрицы с максимальной суммой элементов. В программе реализовано два алгоритма: cadane(matrix) отчасти является реализацией алгоритма, разработанного на конкурсе [**Acceler8**](https://software.intel.com/ru-ru/articles/contest-acceler8-2011-main), проводившегося Intel’ом в 2011 году. В алгоритме была ошибка, которую удалось обнаружить при реализации второго алгоритма, который представлен на тематическом форуме о динамическом программировании. Для экономии ресурсов используется только первый алгоритм. Активировать второй алгоритм можно вызовом функции c\_alg(matr).
7. Обе функции поиска максимальной подматрицы возвращают площадь фигуры и координаты левой верхней точки прямоугольника и нижней правой.
8. Вырезаем изображение
9. Данное изображение увеличивается по оси Y в 1/k раз и уменьшается по оси Х в k раз (шутка для гуманитариев, мы ведь понимаем, что это уменьшает изображение пропорционально в k раз)
10. Результат записывается в файл result.bmp

Программа поддерживает как запуск через IDE (предустановленные параметры работы, требуется tram.jpg), так и запуск с передачей аргументов через консоль. Порядок:

1. Путь к файлу относительно расположения скрипта
2. Целочисленное значение угла поворота (- целое число)
3. Коэффициент - рациональное число
4. Признак используемой интерполяционной схемы (0 – нулевого порядка (linear), 1 – первого порядка(cubic), 2- третьего порядка(area)).

Также я обратил внимание на алгоритм Тадао Такаока, но не смог реализовать его.

**Литература**

1. <https://pypi.python.org/pypi/opencv-python>
2. <https://software.intel.com/ru-ru/articles/maximum_subarray_problem_algorithm_kadane_by_udjin123>
3. <https://www.geeksforgeeks.org/dynamic-programming-set-27-max-sum-rectangle-in-a-2d-matrix/>
4. <https://habrahabr.ru/post/136188/>
5. <https://en.wikipedia.org/wiki/Maximum_subarray_problem#Kadane's_algorithm_(Algorithm_3:_Dynamic_Programming)>
6. <http://program.rin.ru/razdel/html/721.html>
7. http://algolist.manual.ru/maths/linalg/fmaxpminor.php