

Пояснительная записка к тестовому заданию

1. Задача

Необходимо найти кратчайшие пути произвольно заданой точки до всех 4 границ изображения.

- Переход между соседними точками на изображении возможен только по вертикали и горизонтали;
- Стоимость перехода (расстояние) между соседними точками рассчитывается как произвольная функция от цвета соседних точек;
- Длина пути рассчитывается как сумма всех переходов;

Алгоритм решающий поставленную задачу должен быть реализован на C++ и оформлен в виде приложения с удобным UI (выбор точки при помощи мышки, вывод кратчайших путей на изображении и т.п.)

Для чтения изображения и вывода на экран можно использовать любые сторонние кроссплатформенные фреймворки, например OpenCV. Изображение должно быть представлено в формате RGB по 8 бит на канал.

Должны быть реализованы как минимум следующие функции расстояния между точками:

- Сумма модулей разностей по 3-м каналам (RGB);
- Корень из суммы квадратов разностей;
- Сумма яркостей 3-х каналов соседней точки;
- + 1 произвольная на свой выбор;

К программе должна быть приложена небольшая пояснительная записка, описывающая алгоритм, какие алгоритмы рассматривались и почему был выбран именно этот вариант реализации.

1.1. Алгоритм

Для решения поставленной задачи был выбран алгоритм Дейкстры. С его помощью находились длины кратчайших путей до всех точек изображения от указанной. Далее на каждой границе перебором находилась точка с минимальным путем до нее. Затем к каждой из этих точек восстанавливался путь.

Алгоритм Дейкстры позволяет находить кратчайшие пути и их длины во взвешенном графе с неотрицательными весами. Так как предложенные функции расстояния не принимают отрицательных значений, очевидно, что алгоритм Дейкстры подходит для решения данной задачи. Однако существуют другие алгоритмы поиска кратчайших путей на ориентированных графах. Алгоритмы Беллмана-Форда и Левита работают для графов с отрицательными весами. Алгоритмы Флойда — Уоршелла и Джонсона позволяют находить расстояния между всеми парами вершин. Таким образом использование этих алгоритмов излишне для данной задачи.

1.2. Графический интерфейс

Подготовленная программа имеет графический интерфейс.

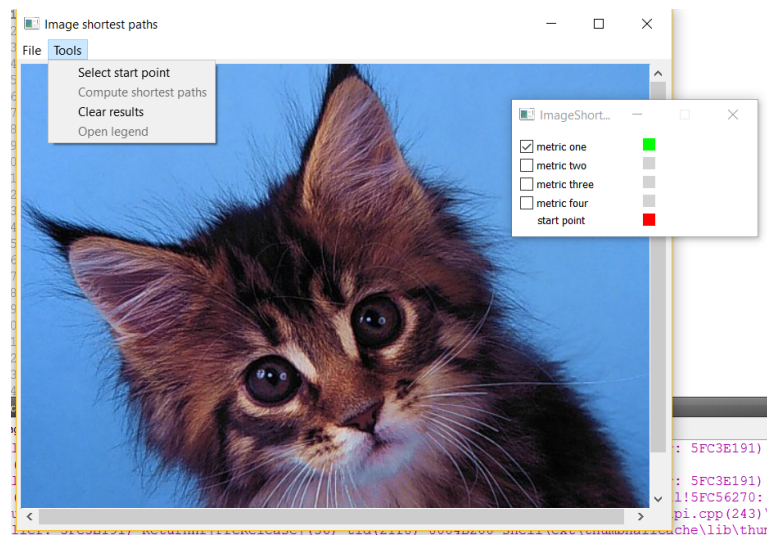


Рис. 1: графический интерфейс пользователя.

Для расчета кратчайших путей необходимо в меню "file" выбрать пункт "open image" и открыть нужную картинку. В меню "tools" выбрать "open legend" и в появившемся окне отметить необходимые метрики и выбрать цвета, которыми будут нарисованы соответствующие им пути. Затем нужно выбрать "select start point" и установить точку на изображении, правой кнопкой мыши можно выйти из режима установки стартовой точки. После этого необходимо нажать "compute shortest paths".