

ПОСТАНОВКА ЗАДАЧИ

Реализовать алгоритм для выделения контура объектов на изображении.

МЕТОД РЕШЕНИЯ ЗАДАЧИ

1. Загрузка и подготовка изображения

Загружаем изображение и преобразовываем его в чёрно–белое (градации серого). Каждый пиксель – значение яркости.

2. Применение маски

Для каждого пикселя изображения берётся область 3×3 пикселей вокруг этого пикселя.

Каждое значение пикселя в этой области умножается на соответствующее значение в маске.

Результаты всех этих умножений суммируются.

Результат этого суммирования становится новым значением пикселя.

$$\text{Фильтр Гаусса: } \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3. Создание бинарного изображения

Все пиксели, чья яркость больше среднего значения (средней яркости изображения) являются фоном (0), а все остальные пиксели (темное порога) — объектами (1). Это создает бинарную маску, где 1 соответствует объекту, а 0 — фону.

4. Поиск контуров

Программа ищет контуры объектов по бинарной маске. Если пиксель объекта (1) хотя бы имеет одного соседа с фоном (0), этот пиксель считается частью контура. Контуры выделяются в отдельном массиве `contours`, где также будут значения 1 для контуров, а для фона — 0.

5. Разделение объектов

Используется метод «волнового алгоритма». Программа обходит все пиксели бинарной маски и, если обнаруживает пиксель объекта (1), который еще не был помечен, она помечает его и проверяет соседние пиксели. Если нашелся пиксель объекта, то помечается таким же номером и добавляется в очередь.

Каждый объект получает уникальный номер. В конце возвращается размеченная маска и количество предметов.

6. Раскрашивание контуров разными цветами

Для каждого объекта по его метке (1, 2, 3, ...) генерируем случайный цвет. Находим все пиксели контура этого объекта.

Если пиксель и контур (`contours[i, j] == 1`), и принадлежит текущему объекту (`labeled_mask[i, j] == label`), красим его в этот цвет.

Чтобы контур был толще, красим не только сам пиксель, но и соседние.