

3 АЛГОРИТМЫ ОБХОДА КОНТУРА

Данные алгоритмы предназначены, например, для последовательного обхода и перекрашивания некоторого контура в другой цвет.

3.1 Простой замкнутый контур

Под *простым замкнутым* контуром понимается любая фигура, не имеющая никаких пересечений, например, окружность, прямоугольник. Для примера возьмем окружность. Если учесть, что экран монитора можно рассматривать как матрицу дискретных пикселей, то любые линии и кривые отображаются на нем в виде набора смежных пикселей (рис.3.1).

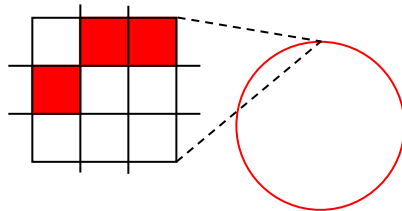


Рисунок 3.1 – Набор смежных пикселей в изображении окружности

Если в качестве условия принять, что точное расположение пикселей не известно, то для того, чтобы обойти простой контур необходимо:

1. Найти любой из пикселей, принадлежащих контуру.

Для этого необходимо последовательно просматривать пиксели экрана, начиная от пикселя $(0, 0)$ до (n, m) (точные значения n и m зависят от графического режима и разрешения монитора), пока не встретится пиксель, принадлежащий контуру, т.е. имеющий исходный цвет контура (рис.3.2). Этот пиксель будем использовать в качестве начальной точки движения по контуру.

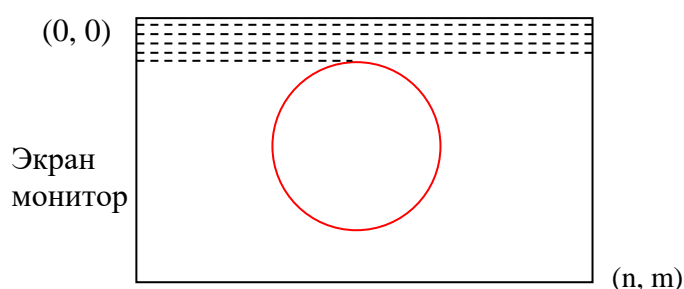


Рисунок 3.2 – Поиск начального пикселя контура

2. Двигаться по контуру.

Для этого необходимо осмотреть окрестности пикселя и перейти в следующий найденный пиксель, принадлежащий контуру (рис.3.3).

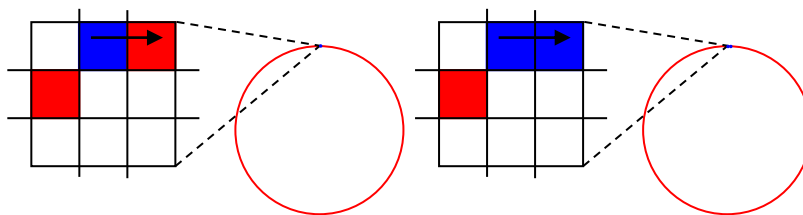


Рисунок 3.3 – Обход контура

3. Остановиться при достижении начальной точки движения.

3.2 Сложный контур

Если контур представляет собой набор пересекающихся линий и кривых, то такой контур называется *сложным*. Опишем алгоритм обхода сложного контура:

1. Найти любой из пикселей, принадлежащих контуру.
2. Двигаться по контуру до тех пор, пока не встретиться пиксель, в окрестностях которого более одного пикселя, принадлежащего контуру.
3. Запомнить координаты пикселя в стек и продолжить движение.
4. Если в окрестностях пикселя все пиксели перекрашены (т.е. нет ни одного неперекрашенного пикселя), то перейти к пикселю, координаты которого были запомнены в стеке и продолжить движение.
5. Если в окрестностях пикселя нет ни одного неперекрашенного пикселя и стек пуст, остановить выполнение алгоритма.