

Баташова Светлана Сергеевна

Batashova Svetlana Sergeevna

Студент

Student

Балаева Маргарита Олеговна

Balaeva Margarita Olegovna

Студент

Student

Погорелов Дмитрий Александрович

Pogorelov Dmitriy Alexandrovich

Старший Преподаватель

Senior lecturer

Московский Государственный Технический Университет

имени Н.Э. Баумана

Bauman Moscow State Technical University

РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ АЛГОРИТМА СО СПИСКОМ РЕБЕР И ФЛАГОМ

IMPLEMENTATION AND INVESTIGATION OF AN ALGORITHM WITH A LIST OF EDGES AND A FLAG

Аннотация: в данной работе реализован алгоритм заполнения со списком рёбер и флагом, также произведён замер времени выполнения алгоритма.

Ключевые слова: алгоритм, заполнение, время работы алгоритма.

Abstract: in this paper, a filling algorithm with a list of edges and a flag is implemented, and the algorithm execution time is also measured.

Keywords: algorithm, filling, algorithm running time.

Алгоритм заполнения со списком рёбер и флагом [1]

Одной из отличительных черт данного алгоритма – наличие флага, который является признаком расположения пикселя внутри или вне многоугольника.

Сам алгоритм состоит из двух этапов, шагов.

Первый шаг – отрисовывается контур, который ограничивает область. А второй – само заполнение (заполнение областей располагающихся между ограничивающими пикселями).

На первом этапе обрабатываются рёбра многоугольника, путём применения множества сканирующих строк. Производятся вычисления точек пересечения ребра и сканирующей строки (для наибольшей эффективности реализуется итеративный способ, чтобы избежать дополнительных затратных вычислений). После этого вычисляется самый левый пиксель, расположенный справа от точки пересечения.

Таким образом, получается контур фигуры.

На втором этапе осуществляется проход по сканирующим строкам, в процессе которого анализируется положение очередного пикселя. Флаг, равный лжи (False), является признаком того, что пиксель не принадлежит области заполнения, а если он равен истине (True), то наоборот, принадлежит. На основе этого признака пикселю присваивается либо цвет фона, либо цвет многоугольника. В начале программы флаг равен значению лжи (False).

Сам признак меняется, когда очередной пиксель имеет граничное значение.

Важный момент

На момент отрисовки контура может возникнуть следующая ситуация: при достаточно малых углах ребра могут “сливаться”, т.е. иметь 1 (или больше, если угол очень маленький по величине) общий пиксель. Это значит, что признак не изменится в нужный момент. Чтобы избежать этой проблемы, нужно в процессе работы проверить эту ситуацию, и если она возникает, то следует высветить соседний пиксель.

Преимущества алгоритма

Одно из главных преимуществ является то, что каждый пиксель обрабатывается только 1 раз (что очень сильно сокращает временные затраты по сравнению с другими алгоритмами).

Кроме того, в этом алгоритме не требуется сортировка рёбер (следовательно, не тратится и время на это).

Реализация

Отрисовка контура фигуры (рисунок 1).

```
edge = edge_list
while edge is not None:
    y = edge.y1
    x = edge.x1

    if edge.y2 == edge.y1:
        continue

    dx = (edge.x2 - edge.x1) / (edge.y2 - edge.y1)
    while y <= edge.y2:
        if (control_pixels(round(x), y) == True):
            draw_pixel(round(x) + 1, y, EDGE)
        draw_pixel(round(x), y, EDGE)

        x += dx
        y += 1
    edge = edge.next
```

Рисунок 1 – отрисовка контура фигуры

Заполнение (перед этим были вычислены значения \min_x/\min_y , \max_x/\max_y – параметры прямоугольной оболочки) (рисунок 2).

```

y = min_y
while (y <= max_y):
    flag = False
    x = min_x
    while x <= max_x:
        if (check_color(x, y) != None):
            flag = not flag
        if (flag == True):
            draw_pixel(x, y, FIG)
        else:
            draw_pixel(x, y, BG)
        x += 1
    if (delay != 0):
        QtWidgets.QApplication.processEvents()
        QThread.msleep(delay)
    y += 1

```

Рисунок 2 – заполнение

В конце условие для задержек, которые можно задать в начале работы; при задержке, равной нулю, фигура будет отрисовываться в обычном режиме.

Анализ времени

Производится анализ зависимости времени работы алгоритма от площади заданной фигуры. В качестве примера был взят квадрат. По оси ОХ указано значение площади (рисунок 3).

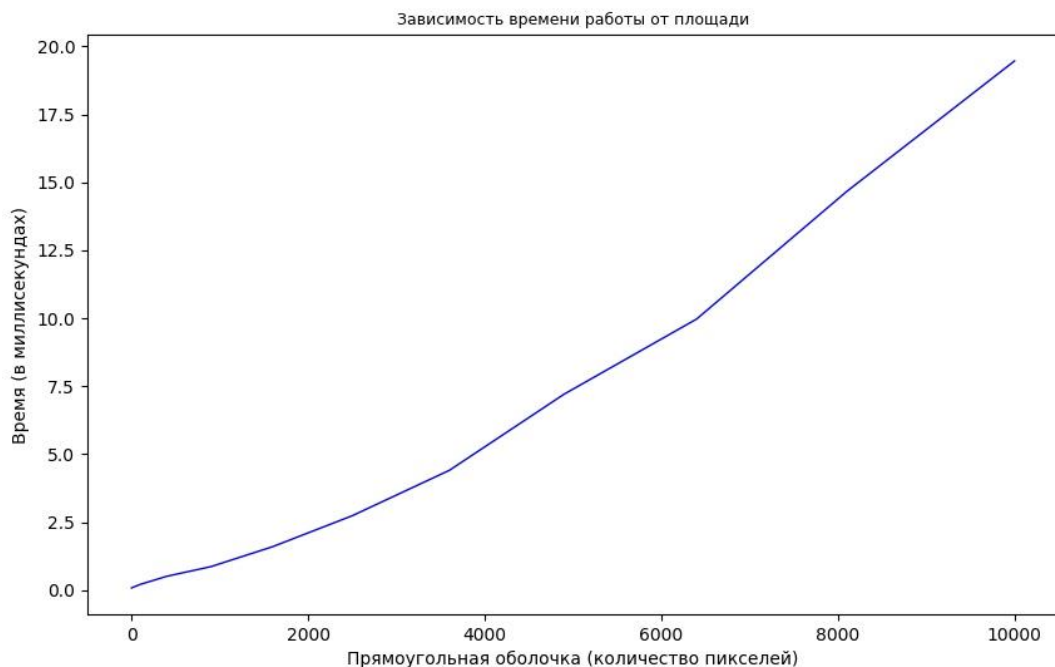


Рисунок 3 – зависимость времени работы от площади

На основе этого графика можно сделать вывод о том, что время работы линейно зависит от площади фигуры.

Список использованных источников

1. ACKLAND B. D., WESTE N.: The edge flag algorithm - a fill method for raster scan displays. IEEE Trans. Computers 30, 1 (1981), 41–48.
2. FOLEY J. D., VAN DAM A., FEINER S. K., HUGHES J. F.: Computer graphics: principles and practice (2nd ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.