

Лабораторные работы

4. Вычислительная геометрия на плоскости. Примечания для алгоритмов, работающих с полигонами. Исходные данные должны читаться из текстового файла, имя которого передаётся в командной строке. Файл представляет собой набор целых чисел, разделенных пробелами или символом перевода строки. Для задания множества точек или контура полигона это количество точек и набор пар координат (x и y), направление контура – произвольное (заранее неизвестно). Кол-во точек не более 100, координаты в пределах от 0 до 400. Например, для квадрата с диагональю (50,70)-(100,120) это следующий набор чисел: 4 50 70 50 120 100 120 100 70.
- 4.1. Отсечение отрезка, алгоритм Сазерленда-Козна. Требуется отобразить прямоугольное окно, по которому будет производиться отсечение. Оно должно иметь центр, совпадающий с центром основного окна и стороны, в два раза меньшие сторон основного окна. Отсекаемый отрезок должен отображаться двумя разными цветами, одним - внутри отсекающего окна, другим - снаружи. Координаты вершин отрезка выбираются случайным образом внутри основного окна (не окна отсечения), по нажатию клавиши программа должна выбирать другой набор координат.
- 4.2. Отсечение отрезка, алгоритм Кируса-Бэка с расчётом промежуточных точек пересечения. Требования к программе см. 4.1.
- 4.3. Отсечение отрезка, алгоритм Кируса-Бэка без расчёта промежуточных точек пересечения (параметрический). Требования к программе см. 4.1.
- 4.4. Отсечение отрезка, алгоритм половинного деления. Требования к программе см. 4.1.
- 4.5. Отсечение отрезка произвольным выпуклым окном, алгоритм Кируса-Бэка с расчётом промежуточных точек пересечения. Кол-во и координаты вершин окна читаются из файла. Координаты вершин отрезка выбираются случайным образом внутри основного окна (не окна отсечения), по нажатию клавиши программа должна выбирать другой набор координат.
- 4.6. Отсечение отрезка произвольным выпуклым окном, алгоритм Кируса-Бэка без расчёта промежуточных точек пересечения (параметрический). Требования к программе см. 4.5.
- 4.7. Определение выпуклости полигона. Кол-во и координаты вершин полигона читаются из файла. В качестве результата на экран выводятся контур полигона сообщение о его виде (выпуклый/невыпуклый).
- 4.8. Вычисление площади выпуклой оболочки. Кол-во и координаты точек читаются из файла. В качестве результата на экран выводятся все точки, выпуклая оболочка и её площадь.
- 4.9. Вычисление периметра выпуклой оболочки. Кол-во и координаты точек читаются из файла. В качестве результата на экран выводятся все точки, выпуклая оболочка и её периметр.
- 4.10. Определение положения точки относительно выпуклого полигона. Кол-во и координаты вершин полигона читаются из файла. Точка выбирается случайно, по нажатию клавиши выбираются другие случайные координаты точки. В качестве результата на экран выводятся контур полигона, положение точки и сообщение о ее положении (внутри/вне/на границе).
- 4.11. Определение положения точки относительно произвольного полигона. Кол-во и координаты вершин полигона читаются из файла. Точка выбирается случайно, по нажатию клавиши выбираются другие случайные координаты точки. В качестве результата на экран выводятся контур полигона, положение точки и сообщение о ее положении (внутри/вне/на границе).
- 4.12. Определение площади общей части двух выпуклых полигонов. Кол-во и координаты вершин обоих полигонов читаются из файла. В качестве результата на экран выводятся контуры начальных полигонов (синий и зеленый), а затем, по нажатию клавиши, в дополнение к ним выводится контур полигона общей части (красный) и его площадь.
- 4.13. Определение периметра общей части двух выпуклых полигонов. Кол-во и координаты вершин обоих полигонов читаются из файла. В качестве результата на экран выводятся контуры начальных полигонов (синий и зеленый), а затем, по нажатию клавиши, в дополнение к ним выводится контур полигона общей части (красный) и его периметр.
- 4.14. Вычитание одного выпуклого полигона из другого. Кол-во и координаты вершин обоих полигонов читаются из файла. В качестве результата на экран выводятся контуры начальных полигонов (синий и зеленый), а затем, по нажатию клавиши, в дополнение к ним, выводятся контуры результирующих полигонов (красный).
- 4.15. Определить площадь объединения (операция “ИЛИ”) двух выпуклых полигонов. Кол-во и координаты вершин обоих полигонов читаются из файла. В качестве результата на экран выводятся контуры начальных полигонов (синий и зеленый), а затем, по нажатию клавиши, в дополнение к ним, выводятся контуры полигонов объединения (красный) и его площадь.

5. Обработка изображения. В качестве исходных данных выступает файл изображения. Его имя передается программе в командной строке как параметр (program.exe image. bmp). Пример изображения в формате BMP прилагается. Программа должна вывести на экран исходное и обработанное изображения рядом.
 - 5.1. Размывающий линейный фильтр 3x3.
 - 5.2. Линейный фильтр 3x3 для повышения резкости.
 - 5.3. Линейный фильтр 3x3 для выделения контуров.
 - 5.4. Линейный фильтр 3x3 для тиснения.
 - 5.5. Псевдотонирование изображения алгоритмом Флойда-Стейнберга. Кол-во битов на цвет изменяется клавишами от 1 до 8.
 - 5.6. Псевдотонирование изображения алгоритмом Байера. Кол-во битов на цвет изменяется клавишами от 1 до 8.
 - 5.7. Приведение изображения к режиму индексированных цветов (построение палитры методом медианного сечения). Кол-во цветов в палитре – 256.

6. Обработка двоичных изображений. В качестве исходных данных выступает файл изображения. Его имя передается программе в командной строке как параметр (program.exe image.bmp). Пример изображения в формате BMP прилагается. Программа должна привести изображение к двоичному виду, пиксел считается черным, если его цвет равен (0,0,0), в остальных случаях он считается белым, вывести на экран исходное (двоичное) и обработанное изображения рядом. Пиксели изображений для наглядности лучше выводить на экран в увеличенном виде (2x2 или 4x4 обычных).
 - 6.1. Определение 8-ми связного контура изображения и сохранение его с помощью кодов направления. Необходимо высветить пиксели контура, построенного по кодам направления, на обработанном изображении другим цветом (например, красным). А также вывести кол-во пикселей в контуре и кол-во битов, необходимое для его хранения.
 - 6.2. Определение 4-х связного контура изображения и сохранение его с помощью кодов направления. Необходимо высветить пиксели контура, построенного по кодам направления, на обработанном изображении другим цветом (например, красным). А также вывести кол-во пикселей в контуре и кол-во битов, необходимое для его хранения.
 - 6.3. Определение 8-ми связного контура изображения и сохранение его с помощью дифференциальных кодов. Необходимо высветить пиксели контура, построенного по кодам направления, на обработанном изображении другим цветом (например, красным). А также вывести кол-во пикселей в контуре и кол-во битов, необходимое для его хранения.
 - 6.4. Определение 4-х связного контура изображения и сохранение его с помощью дифференциальных кодов. Необходимо высветить пиксели контура, построенного по кодам направления, на обработанном изображении другим цветом (например, красным). А также вывести кол-во пикселей в контуре и кол-во битов, необходимое для его хранения.
 - 6.5. Выделить контур изображения, и аппроксимировать его набором отрезков. Максимальное отклонение центра пиксела контура от аппроксимирующего отрезка равно 5 пикселям. Необходимо нарисовать аппроксимирующие отрезки на обработанном изображении другим цветом (например, красным).
 - 6.6. Определение скелета. Необходимо высветить пиксели скелета на обработанном изображении другим цветом (например, красным).

Оформление отчета

В отчет входят:

- титульный лист
- задание
- краткое словесное описание алгоритма
- листинг программы, или той ее части, в которой реализован алгоритм
- результаты работы программы

Даты сдачи отчетов

1. 25 марта 2011
2. 22 апреля 2011
3. 20 мая 2011