

Отчет о практике по программированию  
ученика ... класса школы №57 ... ..

Тема: «Построение диаграммы Вороного  
и триангуляции Делоне множества точек»

**Руководитель практики: Р.В. Гусарев**

**Оценка:**

Сентябрь 2013 года

## 1. Постановка задачи.

Поставленная задача – построить диаграмму Вороного и триангуляцию Делоне заданного множества точек.

Ячейка Вороного данной точки множества есть множество точек, для которых расстояние до данной меньше, чем до любой другой точки исходного множества. Диаграмма Вороного – совокупность ячеек Вороного всех точек множества.

Триангуляция Делоне – триангуляция множества точек, для которой выполнены следующие условия: триангуляция выпуклая, в любой окружности описанной около треугольника триангуляции лежат только вершины данного треугольника. Данная триангуляция обладает рядом экстремальных свойств, например, минимальность максимального угла треугольника в триангуляции. Еще одно ее свойство – двойственность диаграмме Вороного: отрезок АВ входит в данную триангуляцию тогда и только тогда, когда ячейки Вороного точек А и В граничат между собой. Основываясь на этом свойстве мы и будем строить триангуляцию Делоне по вычисленной диаграмме Вороного.

## 2. Метод решения.

Для решения задачи использовался так называемый «алгоритм Форчуна».

Введем вспомогательный объект – заметающую прямую, которая будет проходить через наше множество точек. Будем рассматривать геометрическое место точек, равноудаленных от нашей прямой и точек множества. Как известно ГМТ, равноудаленных от данной точки и данной прямой, есть парабола. Так что исследуемый объект представляет собой объединение дуг разных парабол. Назовем совокупность дуг, наиболее приближенных к нашей прямой, береговой линией. Заметим, что любая граница между дугами береговой линии принадлежит какому-то ребру диаграммы Вороного, так как она равноудалена от каких-то двух точек, но при этом находится не ближе к какой-либо третьей точке, чем к нашей прямой, а значит и к данным точкам (потому что иначе она была бы над береговой линией). Некоторые дуги могут исчезать из береговой линии, при этом две соседние границы сходятся, то есть точка исчезновения дуги из береговой линии есть вершина диаграммы Вороного.

Заметим, что появление новых дуг в береговой линии связано только с прохождением нашей прямой новой точки. Парабола, соответствующая ей (изначально вырожденная в луч), пересекает параболу над собой и образует новую дугу.

Исчезновение же дуги тоже происходит только в определенных местах. Если одна из дуг исчезла, то точка исчезновения есть пересечение трех парабол, значит, эта точка равноудалена от трех точек, соответствующих данным параболом, и нашей прямой, значит, прямая сейчас находится в нижней точке окружности, описанной около данных трех точек. Эту нижнюю точку назовем событием круга.

Таким образом, наша прямая должна побывать в конечном количестве положений, а именно: всех точках множества и во всех образующихся по пути событиях круга, причем последовательно. Так мы сможем отследить все вершины и ребра диаграммы Вороного. При этом для каждого добавленного ребра фиксируются точки, соответствующие данной границе. Так что восстановление триангуляции Делоне после построения диаграммы не составляет сложностей.

## 3. Внешняя спецификация.

Требований к операционной системе нет, так как программа была написана в кроссплатформенном Qt. Но тестирование было проведено только на Windows 7.

## 4. Руководство пользователя.

Программа запускается путем запуска .exe файла в папке с программой. Для добавления

точки на поле требуется кликнуть по полю в месте, где должна быть добавлена точка. Для удаления точек с поля существуют кнопки DELETE LAST POINT (удаляет последнюю добавленную точку) и DELETE ALL POINTS (удаляет все точки на поле). Кнопки DRAW VORONOI и DRAW DELAUNAY включают/выключают отображение диаграммы Вороного и триангуляции Делоне соответственно.

5. Внутренняя спецификация.

Функция `do_voronoi` по имеющемуся очередям точек и событий круга определяет самое приоритетное событие и обрабатывает его.

Функция `process_point` обрабатывает событие точки. Она находит дугу, пересекаемую новой параболой, разделяет ее на две части и между этими частями вставляет новую дугу. Также начинает два новых ребра в точке пересечения новой параболы с найденной дугой.

Функция `process_event` обрабатывает событие круга. Она удаляет соответствующую дугу и начинает на ее месте новое ребро, заканчивая при этом два сошедшихся ребра. После этого проверяется возможность события круга для вновь образованных троек последовательных дуг.

Функция `front_insert` непосредственно осуществляет действия, описанные в `process_point`, при этом проверяет появление событий круга.

Функция `check_circle_event` непосредственно реализует проверку трех дуг на образование события круга.

Функция `circle` вычисляет координату возможного события круга.

Функция `intersect` проверяет пересечение параболы, соответствующей данной новой точке, с данной дугой.

Функция `intersection` находит точку пересечения двух данных парабол.

Функция `finish_edges` проводит окончательное структурирование списка ребер диаграммы. Оставшийся код относится уже не к алгоритму, а к оформлению, так что не прокомментирован и не имеет никакого значения.

6. Комментированный текст программы.

Исходный код алгоритма приложен к программе.

7. Описание тестов.

Тестирование производилось путем многократной проверки работоспособности программы на случайных множествах точек.