

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного моделирования

Алгоритм Форчуна

Доклад

Направление подготовки 09.03.03 Прикладная информатика

Профиль «Прикладная информатика в компьютерном дизайне»

Обучающийся _____

А.В. Быкова

Руководитель _____ доцент ИМКТ А.С. Кленин

Владивосток 2022

Содержание

1	Введение	3
1.1	Диаграмма Вороного	3
1.2	Суть и назначение	3
1.3	Авторство	3
1.4	История развития	4
1.5	Состояние и предшественники	4
1.6	Перспектива использования	4
2	Метод	5
2.1	Описание метода	5
2.1.1	Заметающая прямая и сайты	5
2.1.2	Событие точки и парабола	5
2.1.3	Береговая линия	5
2.1.4	Событие круга	5
2.1.5	Ребра диаграммы	6
2.2	Пример	7
3	Формальная постановка задачи	8
4	Исследование производительности	9
4.1	Результаты исследования	9
5	Список литературы	10

1 Введение

1.1 Диаграмма Вороного

Формально, диаграмма Вороного это $P = \{p_1, p_2, \dots, p_n\}$ — множество точек на плоскости. Если говорить о ее сути, диаграмма Вороного конечного множества точек P на плоскости представляет собой такое разбиение плоскости, при котором каждая область этого разбиения образует множество точек, более близких к одному из элементов множества P , чем к любому другому элементу множества. Она имеет вид:

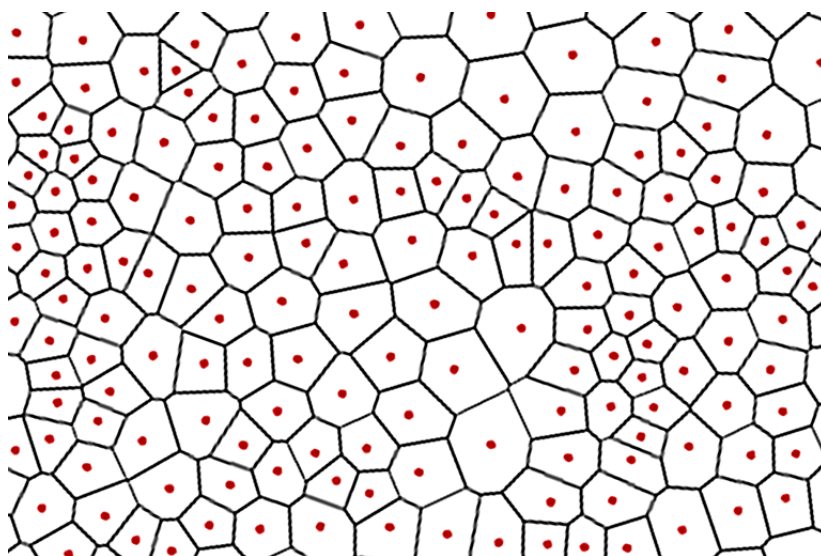


Рис. 1: Диаграмма Вороного

1.2 Суть и назначение

Алгоритм Форчуна – это алгоритм заметающей прямой для генерации диаграммы Вороного из набора точек на плоскости за время $O(n \log n)$ с использованием памяти $O(n)$. Алгоритм берет множество 2D-точек, которые вводятся заранее и строит по ним диаграмму Вороного. Для каждой входной точки, которая называется «сай- том», нам нужно найти множество точек, которые ближе к этому месту, чем ко всем остальным. Такие множества точек образуют ячейки, множество которых и называется диаграммой Вороного.

1.3 Авторство

Данный алгоритм был опубликован Стивеном Форчуном в 1986 году в Нью Джерси во время Второго ежегодного симпозиума по Компьютерной Геометрии. Статья носит название «A Sweepline Algorithm for Voronoi Diagrams» . Название алгоритма происходит от имени его создателя.

1.4 История развития

Алгоритм увидел свет в 1986 году, в статье математика Стивена Форчуна под названием «Алгоритм развертки линий для диаграмм Вороного». Ни один из просмотренных источников не содержит более подробной информации об этом событии, о его предшественниках или об авторе. Так же, отсюда можно сделать вывод, что до Стивена Форчуна никто не интересовался данной темой настолько, чтобы создать полноценный работающий алгоритм, собственно, как и после.

1.5 Состояние и предшественники

На момент своего появления, алгоритм Форчуна был первым в своем роде, который строил диаграмму Вороного с использованием замещающей прямой. Поскольку информация об этом алгоритме ограничена, можно сделать вывод, что он не сыскал популярности в свое время. Возможно это произошло из-за сложности вычислений, а может и совсем по другим причинам.

1.6 Перспектива использования

Алгоритм используется для построения диаграммы Вороного, которая в свою очередь имеет широкий спектр применения. Диаграммы Вороного постоянно использовались антропологами для описания регионов влияния различных культур; кристаллографами для объяснения структуры определенных кристаллов и металлов; экологами для изучения конкуренции между растениями; и экономистами для моделирования рынков в экономике США. Также, она используется в архитектуре, дизайне, поскольку образует красивые причудливые формы.

2 Метод

2.1 Описание метода

2.1.1 Заметающая прямая и сайты

На плоскости находится некоторое количество точек - сайтов. Так же, есть заметающая прямая, которая движется, например, снизу вверх, то есть от сайта с наименьше ординатой к сайту с наибольшей. Вместе с этим, на построение диаграммы влияют только точки, которые находятся ниже или на заметающей прямой.

2.1.2 Событие точки и парабола

Попадание заметающей прямой на очередной сайт вызывает событие точки. В этот момент создается новая парабола, фокусом которой является данный сайт, а директрисой — заметающая прямая.

Эта парабола делит плоскость на две части — внутренняя область параболы соответствует точкам, которые сейчас ближе к сайту, а внешняя область — точкам, которые ближе к заметающей прямой, ну а точки, лежащие на параболе — равноудалены от сайта и заметающей прямой. Парабола будет меняться в зависимости от положения заметающей прямой к сайту — чем дальше она уходит от сайта вверх, тем больше расширяется парабола.

2.1.3 Береговая линия

По мере движения заметающей прямой парабола расширяется, у неё появляются две контрольные точки — точки её пересечения с остальными параболой называются «береговая линия». В береговой линии мы храним дуги парабол от одной точки пересечения их друг с другом до другой, так и получается эта самая линия. Таким образом, данный алгоритм моделирует движение этой береговой линии, где точки пересечения парабол движутся по рёбрам диаграммы Вороного.

2.1.4 Событие круга

В тот момент, когда две точки пересечения — по одной из разных парабол — «встречаются», как бы превращаются в одну, эта точка становится вершиной ячейки Вороного, происходит событие круга. В это время дуга, которая находилась между этими двумя точками — удаляется из береговой линии. Далее мы просто соединяем эту точку с предыдущей подобной ей и получаем ребро диаграммы Вороного. Таким образом происходит построение полноценной диаграммы.

2.1.5 Ребра диаграммы

Ребра диаграммы достраиваются таким образом, что выполняются следующие условия:

- Никакие два ребра не пересекаются (за исключением вершин)
- Каждая вершина (за исключением вершины с наибольшим значением ординаты) непосредственно соединена, по крайней мере, с одной вершиной имеющей большую ординату
- Каждая вершина (за исключением вершины с наименьшим значением ординаты) непосредственно соединена, по крайней мере, с одной вершиной имеющей меньшую ординату

2.2 Пример

Имеется поле с размерами: ширина – 1000, высота – 800. На ввод подаются 5 точек с координатами (208,235), (545,108), (342,601), (724,369), (455,352) соответственно. Программа считывает входные данные, располагает точки в местах, соответствующих их координатам, проходит по ним снизу вверх заметающей прямой и строит диаграмму Вороного. На выход поступает визуальное отображение диаграммы Вороного, подобного вида:

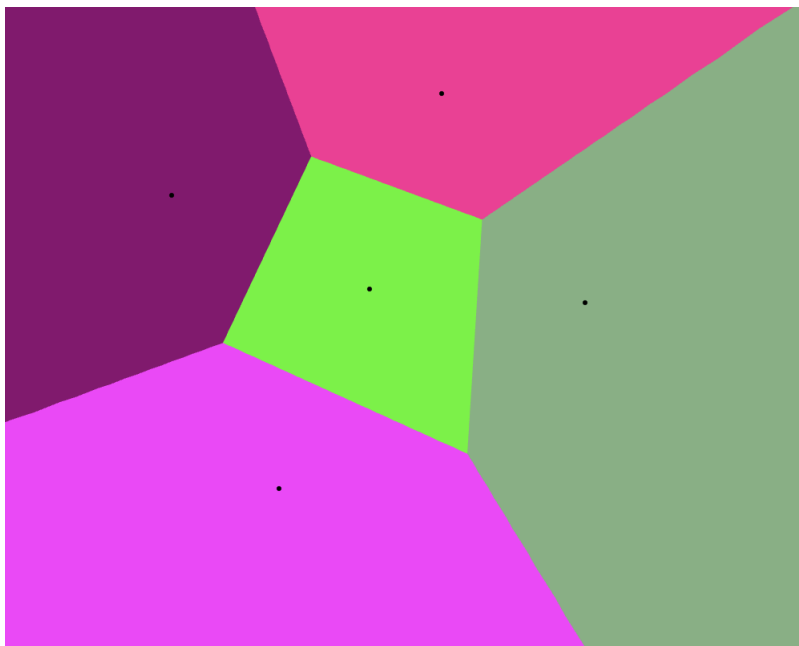


Рис. 2: Пример результата работы алгоритма

3 Формальная постановка задачи

1. Изучить алгоритм Форчуна, описать его в форме научного доклада.
2. Реализовать версию алгоритма Форчуна с его визуализированным изображением. Это должна быть html-страница с визуальным отображением результата работы алгоритма. При заходе на нее должны отображаться изначально заданные 5 точек и диаграмма, построенная по ним. Должен присутствовать список тестовых фигур в виде интерактивных кнопок, при нажатии на которые, отображается соответствующая фигура/рисунок. Также, должна присутствовать кнопка сброса картинки. Формат выходного файла: html-страница, которая содержит диаграмму Вороного.
3. Исследовать алгоритм на предельно допустимое количество входных данных.
4. Результаты работы выложить на гитхаб.

4 Исследование производительности

В рамках доклада была исследована производительность данного алгоритма. Всего были проведены тесты по двум направлениям: по затраченному времени и по количеству потребляемых ресурсов памяти. Исследование проводилось на пяти разных по количеству наборах произвольно взятых точек: 10, 100, 1000, 10000 и 20000, соответственно.

4.1 Результаты исследования

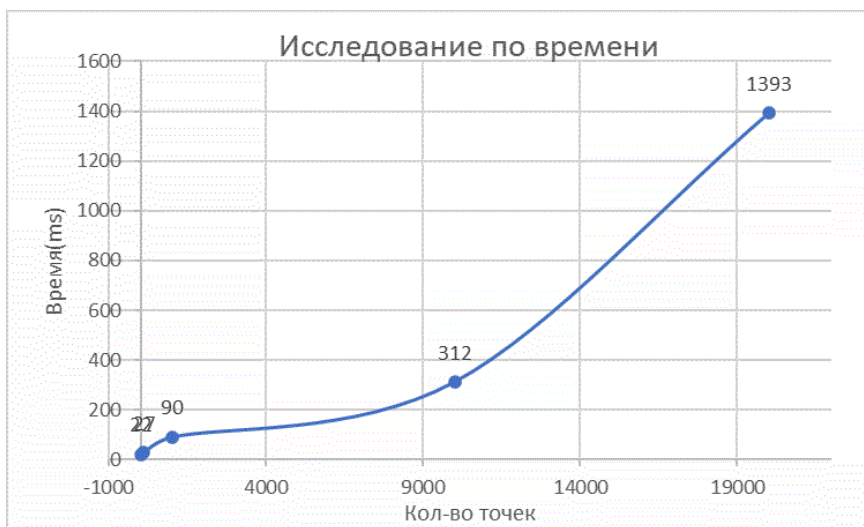


Рис. 3: Время работы алгоритма

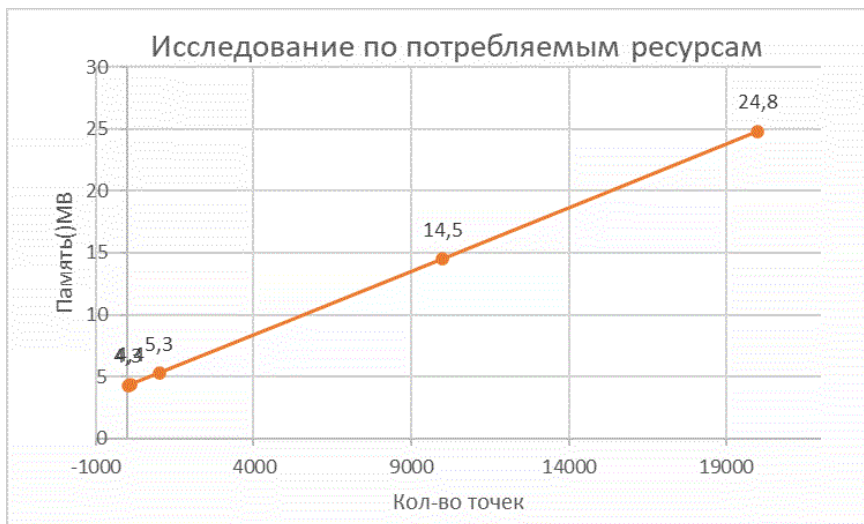


Рис. 4: Затрачиваемый ресурс памяти

5 Список литературы

1. Steven Fortune. A sweepline algorithm for Voronoi diagrams // ACM Digital Library Proceedings of the second annual symposium on Computational geometry. — Yorktown Heights, New York, United States, 1986. — ISBN 0-89791-194-6.
2. Mark de Berg, Marc van Kreveld, Mark Overmars, Otfried Schwarzkopf. Computational Geometry. — 2nd revised. — Springer-Verlag, 2000. — ISBN 3-540-65620-0.
3. David Austin. Voronoi Diagrams and a Day at the Beach. — American Mathematical Society.
4. Rene Descartes, Le Monde, ou Traité de la lumière, Translation and introduction by M.S. Mahoney, Abaris, 1979.
5. A. Okabe, B. Boots, K. Sugihara, S. Chiu, Spatial Tesselations, Concepts and Applications of Voronoi Diagrams, Wiley, 2000.
6. J. O'Rourke, Computational Geometry in C, Cambridge University Press, 2000.
7. Spatial: диаграмма Вороного (Java) [Электронный ресурс] — Режим доступа: <http://obi2ru.blogspot.com/2012/12/spatial-voronoi-diagram-by-java.html>
8. Kenny Wong, Hausi A. Müller, An Efficient Implementation of Fortune's Plane-Sweep Algorithm for Voronoi Diagrams, CiteSeerX 10.1.1.83.5571 .
9. Wikipedia: Алгоритм Форчуна
10. Javascript Implementation of Steven J. Fortune's Algorithm to compute Voronoi diagrams [Электронный ресурс] — Режим доступа: <http://www.raymondhill.net/voronoi/voronoi.html>
11. Алгоритм Форчуна, подробности реализации [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/430628/>
12. Ход «Voronoi» [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/112581/>
13. Ход «Voronoi». Часть 2 — Бинарное дерево [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/112581/>
14. C++ для начинающих. Бинарное дерево. Первое знакомство [Электронный ресурс] — Режим доступа: <https://ci-plus-plus-snachala.ru/?p=89>
15. Практика метапрограммирования на C++: бинарное дерево поиска на этапе компиляции [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/320686/>
16. Fortune's Algorithm in C++ [Электронный ресурс] — Режим доступа: <https://www.cs.mbrubeck/voronoi.html>

17. Voronoi diagrams with Fortune's algorithm [Электронный ресурс] — Режим доступа: <https://www.bitbanging.space/posts/voronoi-diagram-with-fortunes-algorithm>
18. Wikipedia: Fortune's algorithm
19. Fortune's algorithm [Электронный ресурс] — Режим доступа: <https://wikimili.com/en/algorithm>
20. Диаграмма Вороного и её применения [Электронный ресурс] — Режим доступа: <https://itnan.ru/post.php?c=1p=309252>
21. Voronoi Diagrams and a Day at the Beach [Электронный ресурс] — Режим доступа: <https://www.ams.org/publicoutreach/feature-column/fcarc-voronoi>
22. Voronoi diagram in AS3 [Электронный ресурс] — Режим доступа: <https://blog.ivank.net/diagram-in-as3.html>
23. Алгоритм Форчуна на C++ для построения диаграммы Вороного на плоскости [Электронный ресурс] — Режим доступа: <https://www.pvsm.ru/matematika/21>
24. THE BOOST.POLYGON VORONOI LIBRARY [Электронный ресурс] — Режим доступа: <https://www.boost.org/doc/libs/1-52-0/libs/polygon/doc/voronoi-main.htm>
25. Voronoi Diagram using Fortune's Algorithm [Электронный ресурс] — Режим доступа: <https://www.youtube.com/watch?v=dgEt9Go7GvE>
26. Voronoi Diagrams and Procedural Map Generation [Электронный ресурс] — Режим доступа: <https://www.youtube.com/watch?v=3G5d8ob-Lfo>
27. Fortune's Algorithm: The Details [Электронный ресурс] — Режим доступа: <https://pvigier.github.io/2018/11/18/fortune-algorithm-details.html>