Лабораторна робота №9

«Діаграма Вороного – метод Форчуна»

Виконала студентка 3го курсу

Групи ІПС-32

Нортман Юлія

Мова виконання: Java

Оцінка складності:

* Додавання site до берегової лінії – O(log N)
* Сортування N точок по х-координаті – O(N\*log(N))
* Складність алгоритму - O(N\*log(N))

Якщо задано дві точки pi та pj(site), то множина точок ближчих до pi ніж до pj є півплощина(H(pi, pj)), що визначена перпендикулярною до відрізка pipj прямою, яка ділить цей відрізок навпіл і містить в собі точку pi. Множина V(i) точок які розташовані ближче до pi ніж до будь-якої іншої точки утворюється внаслідок перетину N-1 півплощин і являється опуклою багатокутною областю, що має не більше ніж N-1 сторін і називається ***многокутником Вороного*** (cell) точки pi.

Отримані таким чином N областей утворюють розбиття площини і називаються ***діаграмою Вороного***.

**Алгоритм Форчуна**

Алгоритм Форчуна використовує поняття замітаючої прямої та берегової лінії. Замітаюча пряма - горизонтальна пряма, що рухається згори вниз. Берегова лінія – кусково-задана крива, що складається із шматків парабол зверху від замітаючої прямої.

*Визначення параболи*:

Візьмемо точку(назвемо її фокусом) та горизонтальну пряму(назвемо її директрисою), тоді всі точки на площині, для яких відстань до фокусу буде дорівнювати відстані до найближчої точки на директрисі, будуть утворювати параболу.

Нехай фокус має координати (xf, yf), а директриса задається рівнянням y = yd, тоді рівняння параболи буде мати вигляд

Алгоритм використовує чергу подій, що впорядковані в порядку за часом коли вони зустрінуться на шляху замітаючої прямої. Існує два типи подій: подія, коли на шляху замітаючої прямої трапилась точка з вхідної множини(site) та подія, коли відбувся перетин двох ребер майбутнього многокутника Вороного

Берегова лінія – впорядкована колекція дуг(парабол) та ребер між ними, впорядкованих за абсцисою.

В загальному вигляді алгоритм Форчуна буде мати вигляд:

Fill the event queue with site events for each input site.

While the event queue still has items in it:

If the next event on the queue is a site event:

Add the new site to the beachline

Otherwise it must be an edge-intersection event:

Remove the squeezed cell from the beachline

Cleanup any remaining intermediate state

*Обробка події, коли на шляху замітючої прямої зустрівся site*

По-перше, потрібно визначити під якою вже існуючою дугою буде знаходитись нова парабола. Це зробити досить нескладно якщо взяти до уваги той факт, що у цей момент замітаюча пряма проходить через новий site і тому новостворена парабола буде просто вертикально напрямленим променем, що задається лише координатою х.

Комірка, яку утворила нова парабола та комірка, яку утворює знайдена дуга параболи будуть сусідніми у діаграмі Вороного, тому між ними має бути ребро. В процесі виконання алгоритму ми зустрінемо два типи ребер: ребро, яке ще продовжує зростати(назвемо його напівребром) та вже готові ребра.

Використані джерела:

* <https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A4%D0%BE%D1%80%D1%87%D1%83%D0%BD%D0%B0>
* <https://jacquesheunis.com/post/fortunes-algorithm/>
* <http://blog.ivank.net/fortunes-algorithm-and-implementation.html>