УДК 519.688

К.т.н., доцент Сулема Є.С., студент Москаленко В.Ю.

Національний технічний університет України «Київський політехнічний інститут»

МОДИФІКОВАНИЙ ІТЕРАЦІЙНИЙ АЛГОРИТМ ПОБУДОВИ ТРІАНГУЛЯЦІЇ ДЕЛОНЕ НА ПЛОЩИНІ

Вступ

Задача побудови тріангуляції Делоне є однією з базових в обчислювальній геометрії. До неї зводиться чимало інших задач, вона широко використовується в машинній графіці та геоінформаційних системах для моделювання поверхонь та вирішення просторових задач.

Існує багато алгоритмів, які дозволяють побудувати тріангуляцію Делоне на будь-якій поверхні за скінченну кількість кроків [1-4]. В даній статті представлена модифікація існуючого ітераційного алгоритму, яка дозволяє збільшити швидкість побудови тріангуляції.

Постановка задачі

Задача полягає в модифікації існуючого ітераційного алгоритму [4] таким чином, щоб його реалізація була не складнішою, ніж для існуючих алгоритмів, але при цьому швидкість тріангуляції поверхні була б якнайвищою.

Термінологія

Тріангуляція – це планарне розбиття площини на M фігур, з яких одна називається нескінченністю, а інші – трикутниками [1].

Задачею побудови тріангуляції за заданим набором двовимірних точок називається задача поєднання заданих точок відрізками, що не перетинаються, таким чином, щоб в отриманій тріангуляції між будь-якими двома даними точками не можна було провести нові відрізки без перетину з вже існуючими [1].

Тріангуляція називається *тріангуляцією Делоне*, якщо всередину кола, описаного навколо будь-якого з побудованих трикутників, не потрапляє жодна із заданих точок тріангуляції [2].

Опис алгоритму

Усі ітеративні алгоритми передбачають послідовну побудову тріангуляції, під час якої виконується додавання нового трикутника в частково побудовану тріангуляцію. При додаванні точки до вже побудованої тріангуляції виконують такі дії [3]:

1. Визначається, в який трикутник потрапляє дана точка, або знаходиться трикутник на межі тріангуляції, найближчий до даної точки.
2. Проводяться перевірки отриманих трикутників на відповідність умовам тріангуляції Делоне та виконуються необхідні перебудови.

Модифікація існуючого ітеративного алгоритму полягає у зміні способу пошуку трикутника, в який потрапляє точка, або найближчого ребра трикутника, до якого буде під'єднана нова точка:

1. Визначається ребро тріангуляції, яке знаходиться найближче до даної вершини, для цього вираховується відстань від точки до прямої.
2. Визначається, з якої сторони від ребра знаходиться точка, що додається, – ліворуч чи праворуч (обхід трикутників виконується за годинниковою стрілкою).
3. Якщо точка знаходиться ліворуч від найближчого ребра, то ця точка потрапляє всередину трикутника. Наступні дії:

   3.1. Визначається трикутник, в який потрапляє точка. Для всіх вершин цього трикутника нова точка має бути розташована ліворуч. Наступні дії залежать від вибору структури даних:
   - якщо зберігається список базових трикутників, то з нього вибирається трикутник з даним ребром та проводиться перевірка щодо розташування точки в цьому трикутнику відносно його сторін (при цьому враховується, що ребро може входити до двох трикутників);
   - якщо зберігається лише список ребер та список вершин, то виконується пошук найменшого за площею трикутника, якому належить ця точка. Знаходити площу S трикутника доцільно за формулою Герона:

   $$S = \sqrt{p(p-a)(p-b)(p-c)},$$

   де a, b, c – сторони трикутника, p – півпериметр трикутника.

   3.2. Точка додається до знайденого трикутника та створюються три нових ребра.

4. Якщо точка знаходиться праворуч від ребра, то ця точка потрапляє за межі тріангуляції. В цьому випадку створюються нові ребра, які поєднують цю точку із базовими вершинами знайденого ребра.

Після отримання нової тріангуляції необхідно провести перевірку на відповідність умові тріангуляції Делоне. Це можна зробити шляхом перевірки суми протилежних кутів. В [1] показано, що умова Делоне для трикутника з вершинами в точках $A(x_1,y_1)$, $B(x_2,y_2)$, $C(x_3,y_3)$ буде виконуватись тоді і лише тоді, коли для будь-якої іншої точки $D(x_0,y_0)$ тріангуляції буде $\alpha+\beta\le\pi$ (рис.1), тобто

$$\sin\alpha\cos\beta+\cos\alpha\sin\beta\ge 0. \qquad (1)$$

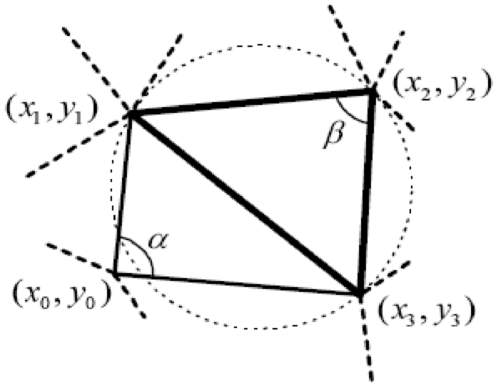

Рис. 1. Перевірка суми протилежних кутів

Значення синусів та косинусів кутів можна вирахувати через скалярні та векторні добутки векторів. Підставивши їх значення в формулу (1) отримуємо наступну умову перевірки:

$$((x_0-x_1)(y_0-y_3)-(x_0-x_3)(y_0-y_1))((x_2-x_1)(x_2-x_3)+(y_2-y_1)$$
$$(y_2-y_3))+((x_0-x_1)(x_0-x_3)+(y_0-y_1)(y_0-y_3))((x_2-x_1)(y_2-y_3)+ \qquad (2)$$
$$+(x_2-x_3)(y_2-y_1))\ge 0$$

Для зменшення кількості обчислювальних операцій спочатку обчислимо часткові значення [1]:

$$S_\alpha=(x_0-x_1)(x_0-x_3)+(y_0-y_1)(y_0-y_3),$$
$$S_\beta=(x_2-x_1)(x_2-x_3)+(y_2-y_1)(y_2-y_3)$$

Умова тріангуляції Делоне виконується лише, якщо $s_\alpha\ge 0$ та $s_\beta\ge 0$, тоді $\alpha\le 90^\circ$ та $\beta\le 90^\circ$; інакше потрібні повні обчислення за формулою (2). Це спрощення дозволяє в середньому на 20-40% скоротити кількість виконуваних арифметичних операцій.

У випадку, коли побудована тріангуляція не є тріангуляцією Делоне, отримані трикутники потрібно перебудовувати відповідно до теореми 1 [4]:

Теорема 1. Тріангуляцію Делоне можна отримати з будь-якої іншої тріангуляції за тією ж самою системою точок, послідовно перебудовуючи пари сусідніх трикутників ABC та ABD, які не задовольняють умову тріангуляції Делоне, в пари трикутників ABD та ACD (рис. 2).

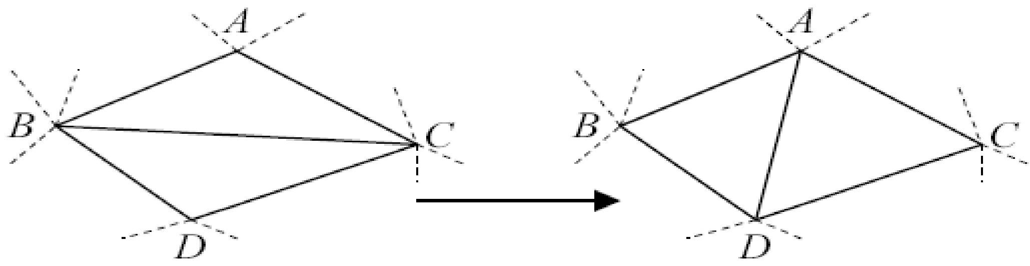

Рис. 2. Перебудова трикутників, які не задовольняють умову Делоне

Запропонований алгоритм повторюють для всіх точок, і, зрештою, отримують тріангуляцію Делоне.

Висновок

При порівнянні запропонованого алгоритму зі звичайним ітераційним алгоритмом отримуємо підвищення швидкості побудови тріангуляції. Наприклад, для 1000 точок алгоритм простої ітерації виконується за час – 0,41 секунд, а модифікований алгоритм – за 0,32 секунд. Обидва алгоритми тестувалися на однакових наборах даних і на одному й тому самому комп’ютері.

Слід зазначити, що у модифікованому алгоритмі зберігається список усіх трикутників. Це дозволяє прискорити пошук чергового трикутника на кожному кроці. Але при цьому значно збільшується об’єм використовуваної оперативної пам’яті. Тому для кожного застосування необхідно вибирати, що важливіше: швидкість чи об’єм оперативної пам’яті.

В подальшому було б доцільно узагальнити цей алгоритм для тривимірних об’єктів.

Література

1. *Скворцов А.В.* Триангуляция Делоне и ее применение – Томск: издательство Томск. ун-та, 2002. – С. 6-74.
2. *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение: Пер. с англ. – М.: Мир, 1989. – С. 16-31.
3. *Скворцов А.В., Костюк Ю.Л.* Эффективные алгоритмы построения триангуляции Делоне // Геоинформатика. Вып. 1. Томск, 1998.– С. 9-47.
4. *Ильман В.М.* Экстремальные свойства триангуляции Делоне // Алгоритмы и программы Вып. 10 (88). М., 1985. – С. 47-66.