УДК 004.92

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ АЛГОРИТМА ФОРЧУНА ДЛЯ ПОБУДОВИ ДІАГРАМИ ВОРОНОГО НА МОВІ ПРОГРАМУВАННЯ PYTHON

ІВАНОВ А. О., (Art_Iv3@ukr.net)
КРИВОНОС О. М. (krypton@zu.edu.ua)
Житомирський державний університет імені Івана Франка

Розглянута поняття діаграми Вороного, а також основні методи її побудови. Це розбиття має широке застосування в географічних інформаційних системах, комп'ютерній графіці, матеріалознавстві тощо, тому необхідними є ефективні алгоритми для його побудови. Запропонована реалізація алгоритму Форчуна для 2-вимірного випадку складністю $O(n^2)$ *мовою програмування Python.*

В багатьох галузях науки виникає необхідність у аналізі та обробці геометрії тих чи інших об'єктів. Одним із потужних інструментів для цього є діаграма Вороного, якій знайшлось широке використання, адже дає можливість, наприклад, ефективно розраховувати короткі відстані, інтерполяції, густини розподілів. Діаграма Вороного являє собою розбиття площини на полігони (в двовимірному випадку) таким чином, що в кожній області є тільки одна точка, яка є найближчою для всіх точок в своїй області. Сама діаграма та її властивості можуть бути використані для визначення властивостей системи, яка розглядається, та вивчення взаємодії її елементів. Також це дає можливість швидко розрахувати тріангуляцію Делоне, яка також має широке використання в розв'язуванні геометричних задач. Існує ряд методів для побудови розбиття Вороного. Їхніми вхідними даними може бути множина точок на площині, а в результаті вони дають представлення полігонів Вороного. У зв'язку з можливими великими об'ємами даних, або їх динамічною зміною, постає проблема розробки та реалізації ефективного та найбільш економного з боку використання ресурсів алгоритму. Це дасть можливість більш детального проєктування та аналізу за рахунок збільшення кількості точок для оброблення.

Алгоритм Форчуна

Одними з найбільш визначними алгоритмами розбиття площини на полігони Вороного є рекурсивний метод (за принципом «розділяй і володарюй»), а також алгоритм Форчуна. Ці методи мають свої переваги і недоліки: перший простий в розумінні, але вимагає врахувати низку нюансів при реалізації; а другий – дає можливість елегантної реалізації, проте потребує попереднього глибокого аналізу та використання більш складних структур даних. Складність обох алгоритмів в найкращій реалізації складає O(n log n). В роботі представлена програмна реалізація алгоритму Форчуна, яка має складність $O(n^2)$ в найгіршому випадку, а також графічний інтерфейс для її використання. Для цього проведено детальний аналіз алгоритму: структур даних і подій, які в ньому реалізуються.

Алгоритм Форчуна базується на понятті «замітаючої прямої» – прямої, яка рухається з одної частини набору точок до іншої, і в залежності від того, які події перетинає ця пряма, виконуються відповідні дії. Таким чином, ми будуємо діаграму Вороного, поступово додаючи до неї нові точки, умовно розділяючи множину точок на три частини: точки, для яких вже побудовані полігони Вороного, точки, які все ще обробляються, та точки, до яких ще не дійшла замітаюча пряма. Також існує 2 типи подій, які визначають зміни в структурах даних, якими ми оперуємо: подія точки (додає до діаграми нову точку), та подія круга (додає новий вузол діаграми). Сам алгоритм Форчуна винесений в окремий файл, який можна імпортувати. В ньому міститься безпосередньо клас самої діаграми Вороного з допоміжними класами.

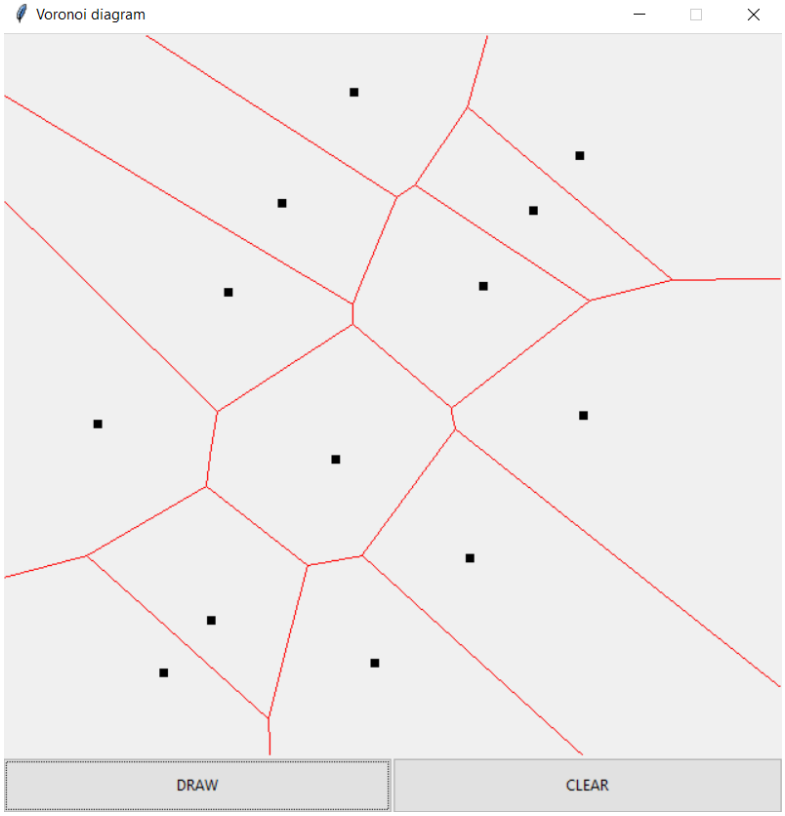

Рисунок 1. Скріншот роботи алгоритму Форчуна для 2-вимірного випадку реалізованого засобами мови програмування Python

Висновки. В результаті виконання роботи було реалізовано алгоритм Форчуна на мові програмування Python. Також були проаналізовані закономірності, які спостерігаються під час проходу замітаючої прямої. Окремо був реалізований графічний інтерфейс (див. рис. 1), який дає можливість тестувати програму. Алгоритм Форчуна є доволі складним, проте крайнє оптимальним з точки зору швидкості і використання пам'яті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, Computational Geometry: algorithms and applications, 2nd rev. ed., Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2000, pp. 147-162
2. W. Pokojski and P. Pokojska, "Voronoi diagrams – inventor, method, applications", *Polish Cartographical Review,* Val. 50, no. 3, pp. 141-150, Oct. 2018, doi:10.2478/pcr-2018-0009
3. S. K. Dey, "Voronoi diagrams in the max-norm: Algorithms, implementation, and applications", PhD thesis, Università della Svizzera Italiana, Lugano, Switzerland, 2015
4. L. Kucera, "Visualization of Abstract Algorithmic Ideas", Available: https://pdfs.semanticscholar.org/2cbb/9df169b20ee81f2af676a32702190fcb2283.pdf
5. O. O. Svitlychnyi and S. V. Plotnytskyi, "Osnovy heoinformatyky: navchalnyi posibnyk [Fundamentals of geoinformatics: a study guide]", Sumy, Ukraine: ВТД "Університетська книга", 2006