Triangulacja Delaunaya

Andrzej Podobiński, Aleksandra Marzec

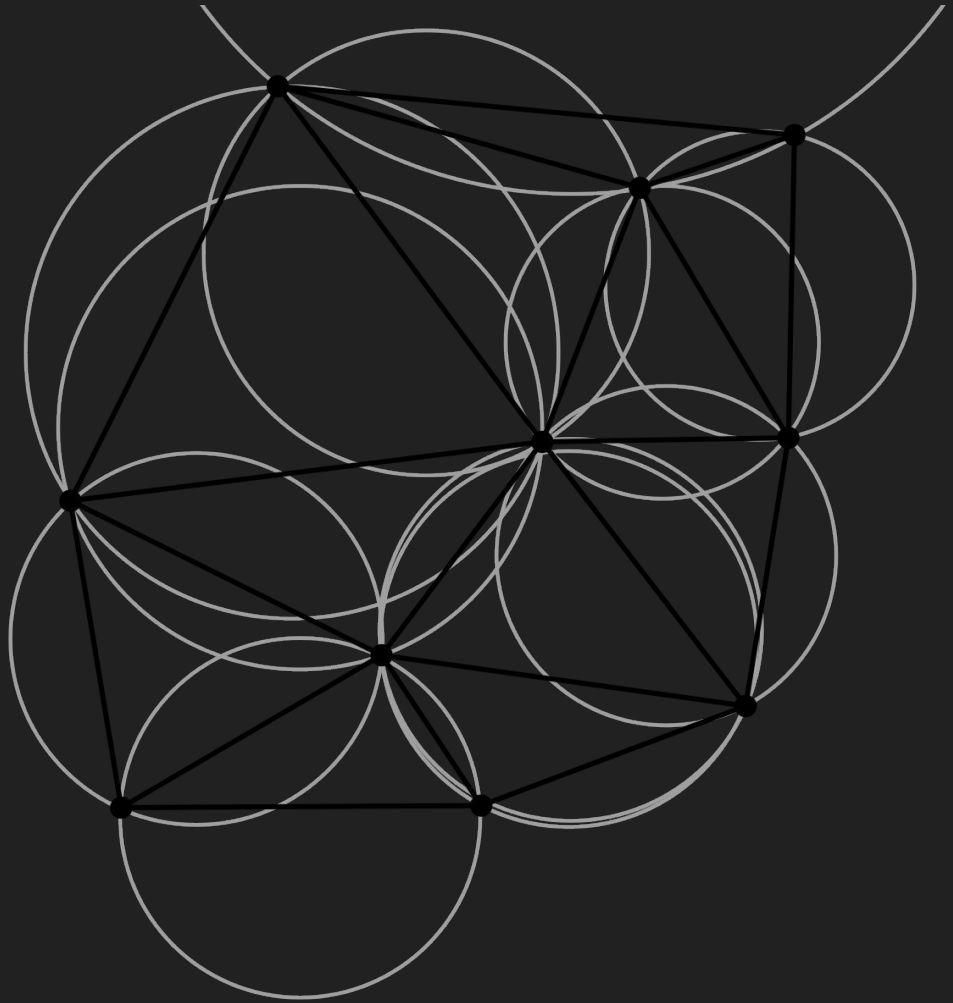
Zadany problem

Mając zadany zbiór punktów 2D, wyznaczyć triangulację Delaunay’a z wykorzystaniem algorytmu iteracyjnego

Przedstawienie problemu

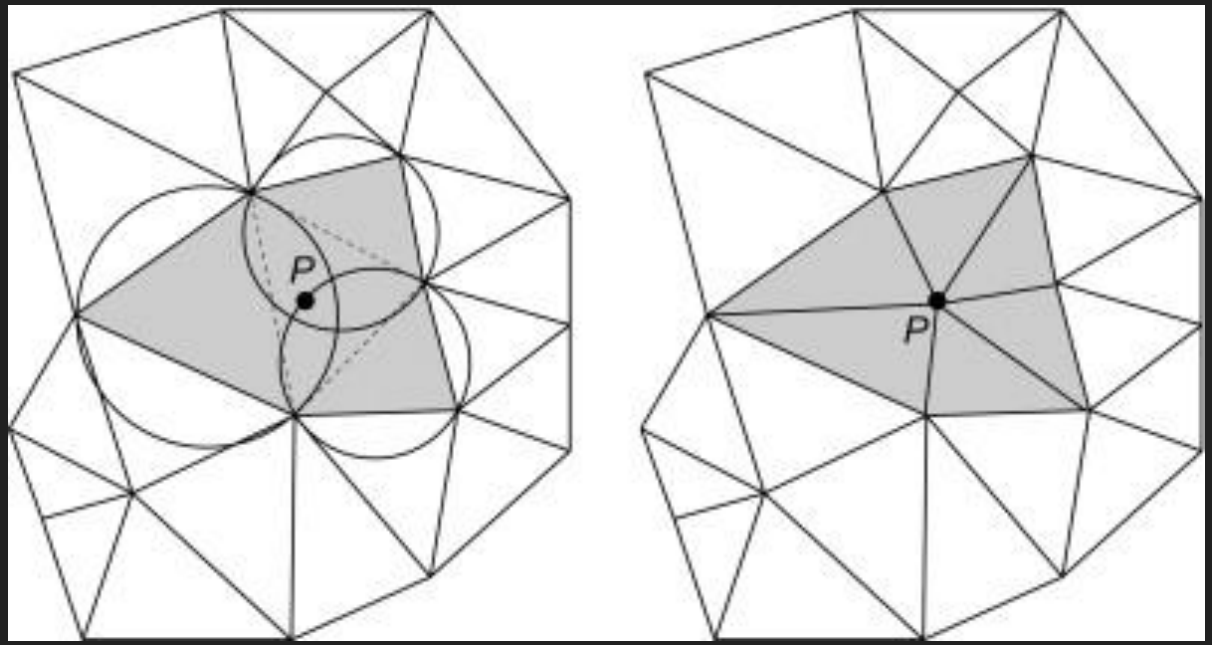
Triangulacja jest podziałem płaszczyzny na trójkąty spełniające warunki: każde dwa trójkąty mają wspólny bok lub nie mają części wspólnej wcale, każdy trójkąt ma część wspólną jedynie że skończenie wieloma trójkątami,

Triangulacja Delanuay-a zbioru punktów jest jednym z rodzajów triangulacji i charakteryzuje się tym, że żaden z punktów tego zbioru nie trafia do wnętrza okręgu opisanego na trójkącie jakiegokolwiek trójkąta powstałego podczas triangulacji



Algorytm iteracyjny (idea Bowyera i Watsona)

- znalezienie prostokąta lub trójkąta zawierającego wszystkie punkty
- początkowa triangulacja T_0 zawierająca jeden lub dwa trójkąty
- dla każdego punktu z chmury punktów , dodanie go do aktualnej triangulacji, wyszukanie trójkątów, których koło opisane zawiera ten punkt
- usunięcie tych trójkątów
- utworzenie nowych elementów poprzez połączenie wierzchołków powstałej wnęki z tym punktem.
- po wprowadzeniu wszystkich punktów, usuwamy wszystkie trójkąty których choć jeden wierzchołek należał to początkowej triangulacji T_0
- zwrócenie triangulacji Delaunay-a


P
P

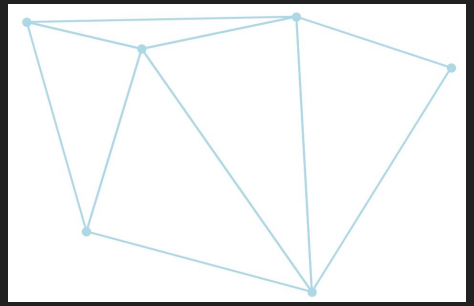
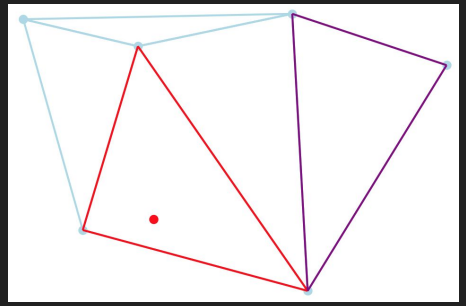
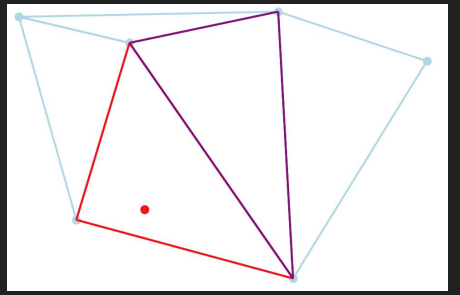
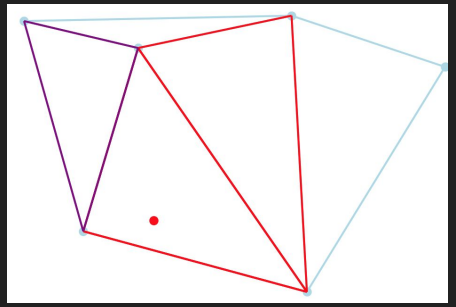
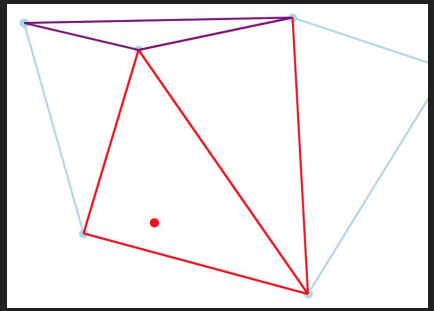
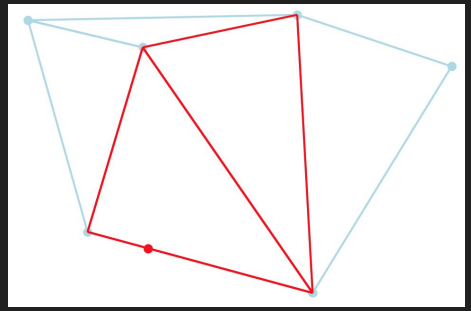
Lokalizacja trójkąta zawierającego dodany punkt

Podstawowym krokiem algorytmu, wpływającym na złożoność jest lokalizacja trójkąta do którego wpada punkt

W naszym algorytmie zaprezentowaliśmy 3 różne metody rozwiązywania tego problemu

Brute Force

Klasycznym sposobem lokalizacji trójkąta jest przeszukanie wszystkich istniejących już trójkątów triangulacji . Zaczynamy od ostatnio modyfikowanego elementu. Jeśli okrąg opisany zawiera dodawany punkt, dodajemy go na listę do usunięcia. Po znalezieniu wszystkich tworzymy nowe elementy .

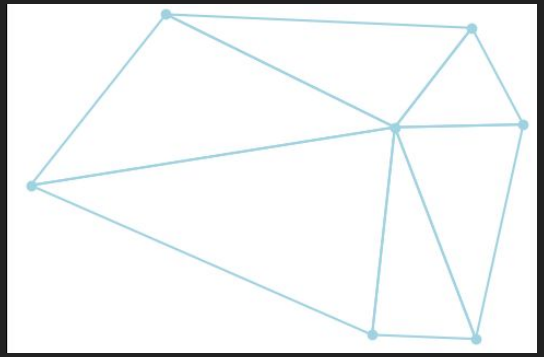
Graf sąsiedztwa topologicznego

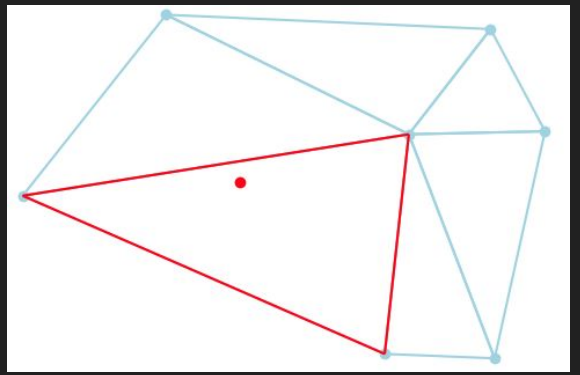
Zoptymalizowany powyższy algorytm polega na skorzystaniu z grafu sąsiedztwa topologicznego. Dla każdego trójkąta i jego krawędzi w słownikach przechowujemy sąsiadów. W przypadku dodawania trójkąta, dla każdej jego krawędzi, jeśli istnieje już w gafie, ustawiamy trójkąt jako nowego sąsiada, jeśli nie , dodajemy krawędź. Podobnie do każdego sąsiadującego trójkąta dodajemy nowego wstawionego sąsiada.

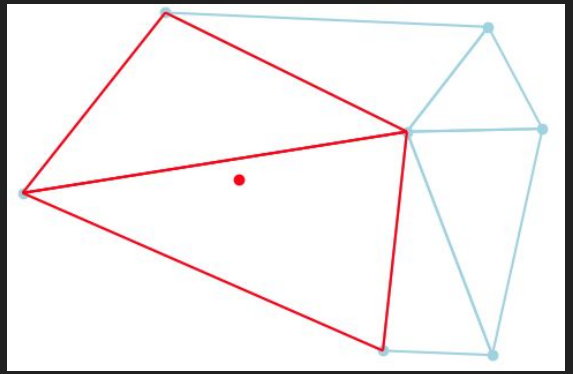
Pierwszy naruszony trójkąt znajdujemy metodą Brute Force, natomiast kolejne z nich szukamy za pomocą stworzonego przez nas iteratora przeszukującego trójkąty triangulacji z użyciem algorytmu BFS(opis niżej)

TriangleIterator

TriangleIterator to iterator stworzony do przeszukiwania zbioru triangulacji w celu znalezienie kolejnego trójkąta do usunięcia (trójkąt, którego okrąg opisany na tym trójkącie zawiera punkt dodany). Korzysta z grafu sąsiedztwa topologicznego, kolejki oraz tablicy visited . Działa na zasadzie algorytmu BFS, jeśli trójkąt nie był jeszcze odwiedzony i punkt wpada do jego okręgu opisanego, dodajemy trójkąt do listy trójkątów do usunięcia. Pomaga to zoptymalizować algorytm poszukiwania trójkąta, ponieważ nie przeszukujemy całego zbioru a jedynie najbliższych sąsiadów

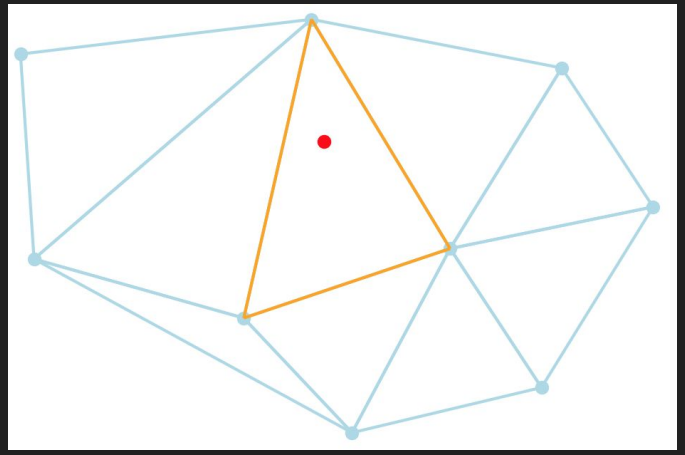


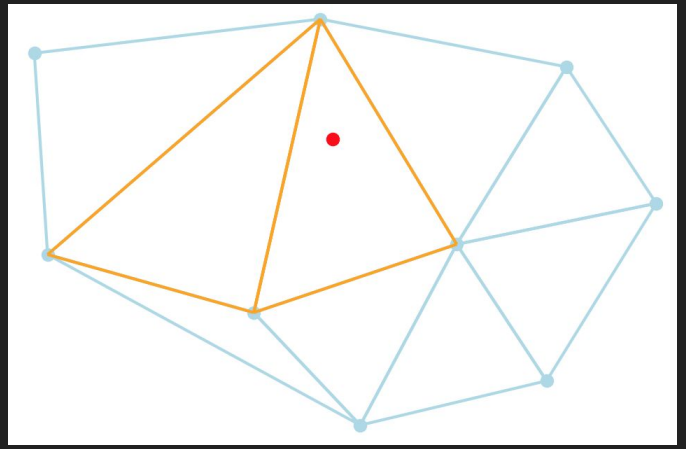


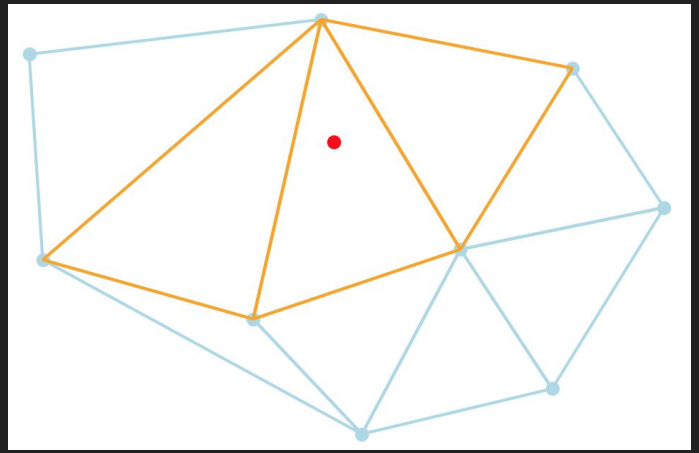


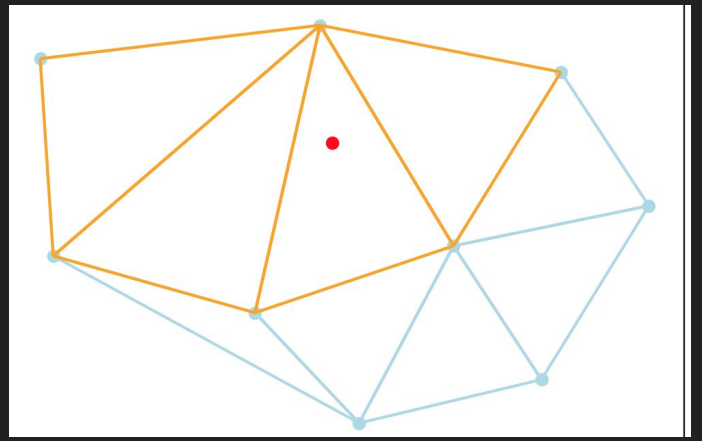
KD-tree

Ostatnim i najbardziej efektywnym sposobem jest użycie KD drzewa z biblioteki. Każdy trójkąt „mapujemy” poprzez środek okręgu opisanego na nim. Struktura kd-drzewa zwraca nam środek okręgu leżącego najbliżej dodanego punktu. Czynność powtarzamy dopóki punkt nie znajdzie się w jakimś trójkącie. Następnie przy użyciu iteratora znajdujemy naruszonych sąsiadów (podobnie jak wyżej).









Porównanie złożoności

Sposób brute force oraz z użyciem grafu sąsiedztwa topologicznego mają taką samą złożoność O(n^2), jednak drugi sposób wypada lepiej ponieważ trójkąt do którego wpada punkt znajdujemy średnio w połowie przeszukiwania.

Sposób z użyciem kd-drzewa ma złożoność O(m*log(n)) dla średniego przypadku, gdzie m to jest liczba trójkątów do których wpada punkt, a n to liczba trójkątów triangulacji.

Bibliografia

1. http://home.agh.edu.pl/~jurczyk/papers/diss.pdf

2. prezentacja z wykładu Triangulacja Delaunay

3. https://en.wikipedia.org/wiki/Delaunay_triangulation

4. https://python-kdtree.readthedocs.io/en/latest/