Lokalizajca punktu - algorytm Kirkpatricka

Patryk Wojtyczek i Agnieszka Dutka

1 Opis Problemu

Na wejściu otrzymujemy spoligonizowaną przestrzeń i punkt P. Zwracamy wielokąt w którym znajduje się punkt P.

2 Działanie algorytmu

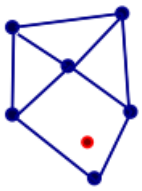
2.1 Wstęp

Algorytm Kirkpatricka zaczyna od zbudowania z wejściowej przestrzeni efektywnej struktury wyszukiwań. Lokalizacja punktu na zbudowanej strukturze jest operacją o złożonści O(logn).

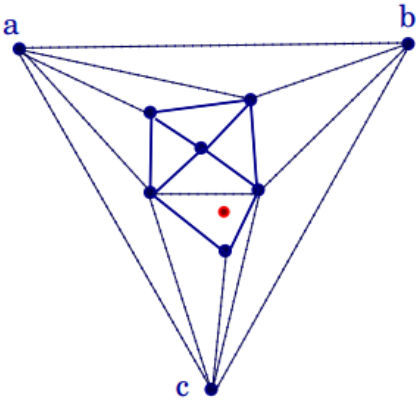
2.2 Preprocessing

Etap I

Tworzenie naszej struktury rozpoczynamy od triangulacji naszego obszaru. Jeśli wejściowy obszar jest striangulowany możemy pominąć ten krok. Wybór triangulacji nie ma znaczenia jako, że ważna jest złożoność wyszukiwań na utworzonej strukturze, a nie czas tworzenia struktury (w naszej implementacji użyliśmy naiwnego algorytmu $O(n^2)$) Następnie szukamy otoczki wypukłej dla naszego obszaru i otaczamy go trójkątem (zupełnie dowolnym). Następnie szukamy otoczki wypukłej i wykonujemy triangulację przestrzeni pomiędzy zewnętrznym trójkątem a otoczką wypukłą.



Rysuenk 1. Wejściowy obszar



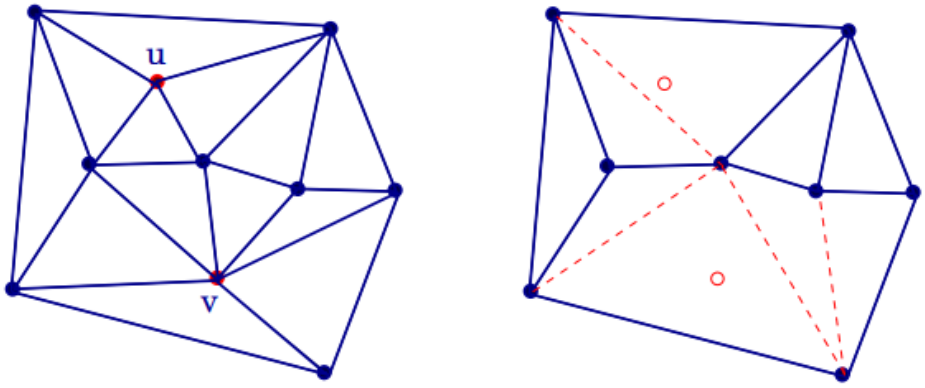
Rysunek 2. Obszar po pierwszym etapie preprocessingu

Etap II

Naszym celem jest "spłaszczenie" triangulacji, którą otrzymaliśmy z poprzedniego etapu do jednego trójkąta (Δabc). Ściślej chcemy wyprodukować sekwencję triangulacji $T_0, T_1, T_2..., T_k$ spełniających własności

1. T_0 to wejściowa triangulacji z Etapu I-ego.
2. T_k to zewnętrzny (Δabc)
3. Liczba triangulacji k jest rzędu O(logn)
4. Każdy trójkąt z triangulacji T_{i+1} przecina niewielką, ograniczoną (O(1)) liczbę trójkątów w triangulacji T_i

Do konstrukcji takiej figury będziemy usuwać wierzchołki i przyległe do nich krawędzie i ponowanie triangulować powstałe w ten sposób dziury. Aby zachować własności wyżej wymienione każdy wierzchołek do usunięcia musi mieć "stały stopień" (O(1)) w grafie T_i. Aby triangulacje na siebie nie zachodziły będziemy wybierać zbiór niezależnych wierzchołków albo inaczej zbiór wierzchołków rozłącznych krawędziowo. W ten sposób powstała dziura z usunięcia wierzchołka może być rozpatrywana lokalnie.

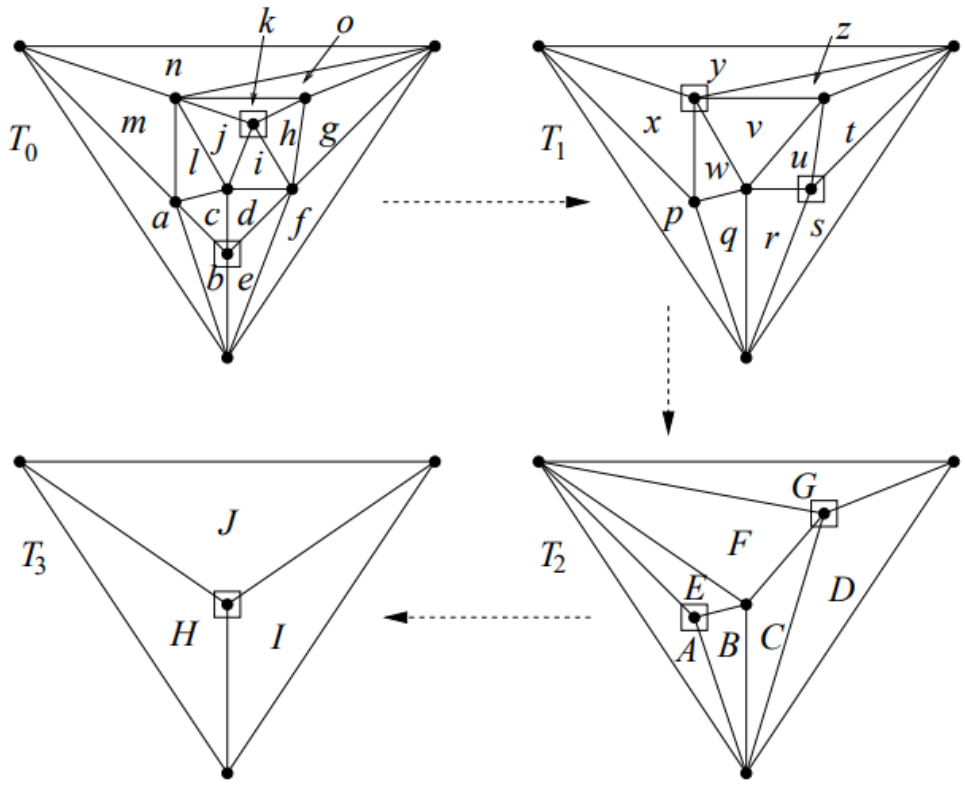


W przykładzie wyżej jako niezależne wierzchołki wybraliśmy u i v, a następnie strianguloważliśmy powstałe przez ich usunięcie dziury.

Do znalezienia zbioru rozłącznych wierzchołków wykorzystujemy naprzemienne kolorowanie grafu przechodząc po nim wszerz.

Procedura postępowania wygląda następująco:

- Znajdź zbiór rozłącznych wierzchołków z grafu T_i
- Usuń znaleziony zbiór z grafu i lokalnie strianguluj powstałe dziury.
- Dla każdego usuwanego trójkąta z dziury zapamiętaj trójkąty z którymi ten trójkąt się przecina po ponownej triangulacji.
- powtarzaj aż triangulacja T_k będzie składała się tylko z trójkąta Δabc



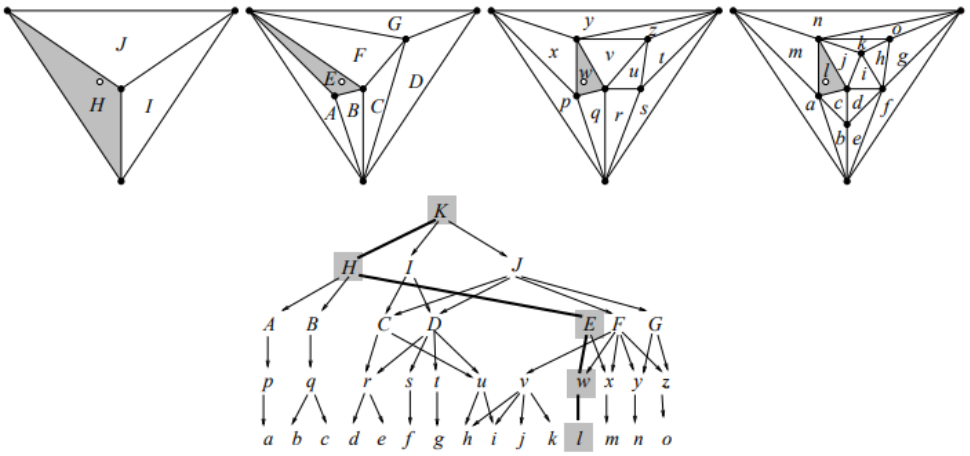
W kwadratach zaznaczamy wierzchołki wybrane do usunięcia.

2.3 Algorytm wyszukiwania

Sprawdź czy p leży wewnątrz T_k (trójkąt Δabc) jeśli nie, zakończ, jeśli tak:

- Przeglądnij dzieci rozpatrywanego trójkąta i ustal do którego należy punkt p.
- powtarzaj aż dojdziesz do T_0

Pamiętając do którego wielokąta należał każdy trójkąt w triangulacji T_0 zwróć ten wielokąt.



3 Dokumentacja

3.1 Opis instalacji i wymagań

1. Projekt został wykonany z Python 3.7
2. Spis bibliotek potrzebnych do uruchomienia projektu znajduje się w pliku requirements.txt
3. Do instalacji najprościej użyć narzędzia pip - wywołaj komendę pip install -r requirements.txt (będąc w katalogu projektu)
4. Wizualizacja została wykonana przy użyciu JupyterNotebook

3.2 Podstawowe klasy

- Point - klasa modelująca punkt na płaszczyźnie 2D.
- Polygon - klasa reprezentująca wielokąt. Zawiera listę punktów zorientowanych przeciwnie do ruchu wskazówek zegara.
- Triangle - klasa dziedziczy po Polygon. Reprezentuje pojedyńczy trójkąt uzyskany w wyniku triangulizacji.
- Kirkpatrick - główna klasa programu - podczas inicjalizacji obiektu tej klasy, tworzy opisaną strukturę poszukiwań.

  Implementuje metodę locate(p: Point), która jest odpowiedzialna za wyszukiwanie punktu w strukturze. Zwraca krotkę zawierającą wielokąt (Polygon) w której znajduje się wyszukiwany punkt albo None jeśli punkt nie znajduje się wewnątrz naszej struktury oraz listę warstw wygenerowanych triangulacji (służy do wizualizaji), które zostały przeglądnięte poprzez przebieg algorytmu.

3.3 Opis użycia

- Utwórz obiekt klasy Kirkpatrick wywowłując konstruktor z listą wielokątów
- Wywołaj metodę locate na utworzonym obiekcie podając szukany punkt.

Przykład
locator = Kirkpatrick(polygons, plot_layers=False)
locator.locate(point, plot_search_=False)

3.4 Wizualizacja

Program umożliwia wizualizację kolejnych etapów dzialania algorytmu

- wyświetlenie kolejnych poziomów wygenerowanego grafu. (przy tworzeniu struktury)

- wyświetlenie znalezionego wielokąta w którym znajduje się punkt.
- wyświetlanie kolejnych rozpatrywanych obszarów przy wyszukiwaniu.

Plik kirkpatrick_visualization zawiera liczne przykłady korzystania z programu i wizualizacji.