

Lokalizacja punktu w przestrzeni dwuwymiarowej – metoda doskonalenia triangulacji – algorytm Kirkpatrick’a

Michał Cyrkowski, Mateusz Wieczorek

Algorytm

Algorytm polega na stworzeniu struktury drzewiastej, w której kolejne poziomy zawierają większą granulację trójkątów, co w efekcie przy pomocy prostego algorytmu podobnego do przeszukiwania binarnego umożliwia przydzielenie punktu do określonego poligonu. Jednak, żeby osiągnąć swoją złożoność $O(n \log h)$ konieczne jest liniowe dzielenie poligonów. Gwarantem złożoności jest prawo Eulera, które stwierdza że w każdym n wierzchołkowym grafie planarnym istnieje co najwyżej $2n - 4$ regiony i $3n - 6$ krawędzi. Dzięki nie dodawaniu nowych wierzchołków złożoność jest ograniczona.

Wykorzystane biblioteki

Do przeprowadzenia triangulacji wykorzystano klasę `Graph` z biblioteki `networkx` oraz klasę `Delaunay` z biblioteki `SciPy`.

Struktury danych

W poniższej tabeli zostały zawarte dowolnie wybrane składowe poszczególnych klas, które wchodzą w skład algorytmu.

PolygonalArea	Klasa wykorzystywana do podzielenia obszaru, na którym będzie pracował algorytm, na większą ilość wielokątów (na początku inicjalizowany jest jeden wielki trójkąt będący granicami obszaru)
points	Tablica współrzędnych punktów wszystkich możliwych wielokątów.
polygons	Tablica definicji wielokątów, z których każda jest tablicą indeksów wskazujących na punkty, z których zbudowany jest dany wielokąt (punkty definiowane w kolejności CCW).

PolygonsTriangulator	Klasa wykorzystywana do dokonania triangulacji na obszarze wykorzystywanym przez algorytm. Każdy zdefiniowany wielokąt jest triangulowany niezależnie.
triangulations	Tablica obliczonych triangulacji dla poszczególnych wielokątów (zgodny z kolejnością definicji). Każda z triangulacji jest tablicą definicji trójkątów, natomiast definicja każdego trójkąta zawiera 3 indeksy wskazujące na elementy-punkty tablicy points, z których zbudowany jest dany trójkąt (punkty definiowane w kolejności CCW).
Triangle	Klasa reprezentująca jeden trójkąt - węzeł w DAG reprezentującym drzewo wyszukiwania trójkątów. Takim trójkątem nie jest tylko trójkąt pochodzący z triangulacji bazowej wielokątów, ale również każdy nowy trójkąt definiowany podczas przetwarzania poszczególnych warstw.
triangleId	Identyfikator trójkąta (wyświetlany numerek w środku trójkąta).
pointsIndices	Definicja trójkąta (analogiczna do tej opisanej w polu triangulations klasy PolygonsTriangulator).
polygonId	Jeżeli jest różna od -1 (tylko i wyłącznie w przypadku liści DAG-u), to zawiera ono indeks wielokąta, do którego należy dany
TrianglesSearchDagGenerator	Klasa wykorzystywana do zbudowania DAG-u reprezentującego drzewo poszukiwań trójkątów.
graphPoints	Zbiór wierzchołków grafu z aktualnego etapu wykonywania algorytmu.
constNodes	Indeksy wierzchołków, które nie mogą zostać usunięte (wierzchołki tworzące wejściowy trójkąt).
trianglesSearchDag	Jeżeli jest różne od None, to zawiera gotowe, zbudowane DAG będące przedmiotem algorytmu.
trianglesSearchDagNodes	Tablica węzłów DAG-u.
currentLayer	Indeksy węzłów wchodzące w skład aktualnie budowanej warstwy DAG-u.
graph	Graf służący do działania algorytmu.
layers	Tablica definicji poszczególnych warstw, każda z tych definicji reprezentuje te same informacje co są zawarte w polu currentLayer.

AbstractNode	Klasa wykorzystywana jako element przechowujący informacje o pojedynczym trójkącie. Jest wykorzystywana podczas przeglądania struktury przypominającej drzewo i stanowi jego węzeł.
getChildren	Zwraca listę trójkątów, które miały wpływ na powstanie pola
getArea	Zwraca trójkąt, który jest reprezentowany przez węzeł
AbstractTriangleTree	Klasa zawierająca informacje o wyniku działania pierwszej części algorytmu. Jest wykorzystywana przy wyszukiwaniu końcowego wyniku
getRootNode	Zwraca korzeń struktury drzewiastej
getNodes	Zwraca wszystkie składowe struktury jako listę

Wizualizacja

Do przeprowadzenia wizualizacji wykorzystano narzędzie graficzne wykorzystywane podczas laboratoriów z przedmiotu. Narzędzie zostało zmodyfikowane, aby umożliwić wizualizację na jednej animacji. Aby zdefiniować dowolny poligon wewnątrz trójkąta należy kliknąć jeden z początkowych punktów trójkąta i następnie zaznaczać kolejne wierzchołki poligonu. Algorytm kończy działanie w chwili kliknięcia już istniejącego punktu. Dodano przyciski trianguluj, który uruchamia algorytm. Następnie w momencie przeszukiwania drzewa w wizualizacji wykorzystano kolory:

- **Zielony** - aktualnie sprawdzany trójkąt
- **Pomarańczowy** - jeszcze nie sprawdzone trójkąty
- **Czerwony** - już odrzucone trójkąty

Bibliografia

1. Wikipedia: https://en.wikipedia.org/wiki/Kirkpatrick%E2%80%93Seidel_algorithm
2. Planar Point Location, Professor Diane Souvaine, Tufts University, Spring 2003: <http://www.cs.tufts.edu/comp/163/notes05/Kirkpatrick.pdf>
3. Dokumentacja networkx: <https://networkx.org/documentation/stable/index.html>
4. Dokumentacja SciPy: <https://docs.scipy.org/doc/scipy/reference/>