Wyznaczanie otoczki wypukłej

Sprawozdanie z laboratorium 13 – Piotr Sarna LK1

Cel ćwiczenia

Podczas zajęć zapoznaliśmy się ze sposobami rozwiązywania problemów geometrycznych. Jednym z nich jest Algorytm Jarvisa.

Wstęp teoretyczny

Algorytm Jarvisa służy do wyznaczenia punktów, które połączone ze sobą stanowią wielokąt okalający wszystkie pozostałe punkty w zbiorze (jest „otoczką wypukłą” zbioru).

Punkty posiadają współrzędne (X, Y) reprezentujące ich położenie na płaszczyźnie.

W algorytmie możemy wykorzystać iloczyn wektorowy, do oszacowania wzajemnego położenia punktów. Dla trzech punktów a, b, c, które tworzą wektory ab oraz ac obliczamy ich iloczyn wektorowy i patrzymy na znak iloczynu:

* > 0 – skręt w lewo, punkt c po lewej stronie
* < 0 – skręt w prawo, punkt c po prawej stronie
* = 0 – punkty leżą na jednej linii

Opis algorytmu

Algorytm rozpoczynamy od znalezienia najniżej położonego punktu z całego zbioru, oznaczamy go jako P0:

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Następnie zaczynając z punktu P0, iteracyjnie wybieramy kolejny punkt, który tworzy największy kąt w prawo względem poprzedniego wektora. Aby porównać, który punkt daje największy skręt, używamy iloczynu wektorowego trzech punktów:

* Poprzedniego punktu r
* Obecnego kandydata pointVector[tempPointIndex]
* Nowego kandydata

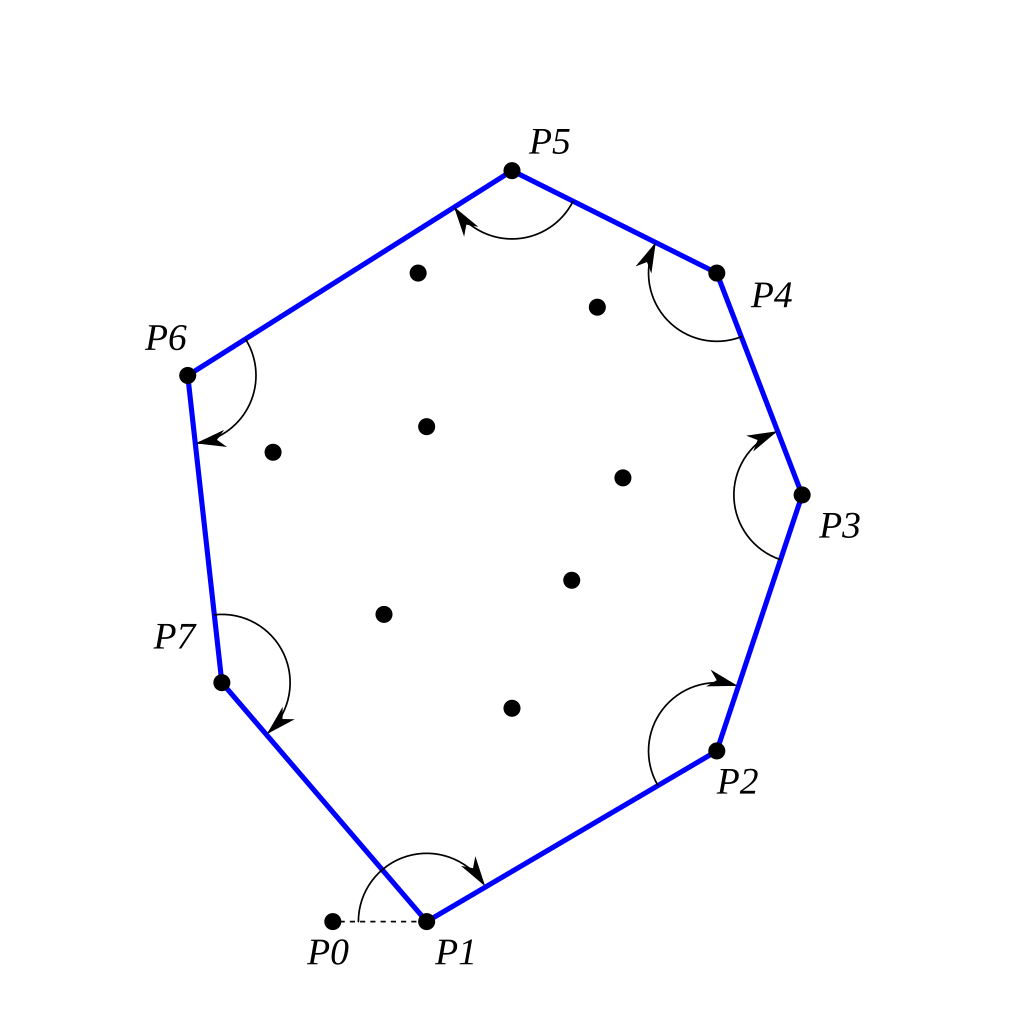
Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Jeżeli iloczyn wektorowy jest mniejszy od zera, oznacza to że nowo badany punkt jest jeszcze mocniej wysunięty, co za tym idzie tworzy jeszcze większy kąt. Aktualizujemy tempPointIndex.

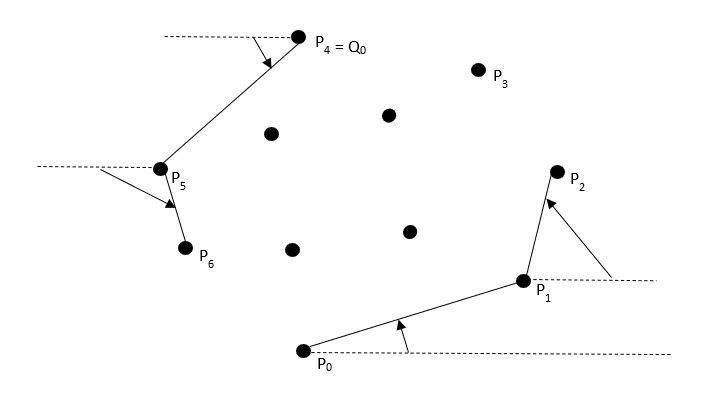
Powtarzamy iterację aż do powrotu do początkowego punktu P0 – oznacza to, że wyznaczyliśmy pełną otoczkę.

Gotową wyznaczoną otoczkę poprzez nasz algorytm można zwizualizować w następujący sposób:



Źródło: <https://pl.wikipedia.org/wiki/Plik:Jarvis_algorithm2.svg>

Alternatywną wersją algorytmu jest wyznaczenie najniżej i najwyżej położonych punktów na płaszczyźnie. Następnie porównując kąty tworzone odpowiednio z dodatnią oraz ujemną półosią OX wyznaczamy największe kąty, Przesuwamy się najpierw z najniższego wierzchołka P0 do najwyższego Q0 a następnie analogicznie w drugą stronę.



To podejście niesie za sobą jednak następujące wady:

* Funkcja atan2(y, x) zwraca wynik w przedziale [-π, π]. Taka komplikacja nie występuje przy podejściu z iloczynem wektorowym.
* Należy osobno rozpatrywać przypadki z punktami współliniowymi – funkcja atan2() zwróci tą samą wartość.

Prezentacja działania implementacji w C++

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, Oprogramowanie graficzne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Wnioski

Algorytm cechuje złożoność obliczeniowa O(n \* h), gdzie n oznacza liczbę punktów zbioru przechowanych w pointVector a h oznacza liczbę punktów otoczki.

Bibliografia

<https://pl.wikipedia.org/wiki/Algorytm_Jarvisa>