Sprawozdanie z Implementacji Algorytmu Triangulacji Wielokątów y-Monotonicznych

Artur Radwański - grupa 4

24 listopada 2025

1 Realizacja Ćwiczenia

1.1 Cel

Celem ćwiczenia była implementacja kluczowych etapów przetwarzania wielokątów w geometrii obliczeniowej: sprawdzenie y-monotoniczności, kategoryzacja wierzchołków oraz właściwa triangulacja wielokąta y-monotonicznego przy użyciu algorytmu opisanego na wykładzie.

1.2 Wstęp teoretyczny

Wielokąt jest **y**-monotoniczny, jeśli każda prosta pozioma przecina go w co najwyżej dwóch punktach. Taki wielokąt można podzielić na dwa łańcuchy z których każdy jest przecinany co najwyżej raz przez każdą poziomą prostą.

Wierzchołki dowolnego wielokąta możemy skategoryzować ze względu na położenie sąsiadów w następujący sposób:

- **Początkowy** - Sąsiedzi poniżej, kąt wypukły ($< 180°$).
- **Końcowy** - Sąsiedzi powyżej, kąt wypukły ($< 180°$).
- **Łączący** - Sąsiedzi powyżej, kąt wklęsły ($> 180°$).
- **Dzielący** - Sąsiedzi poniżej, kąt wklęsły ($> 180°$).
- **Prawidłowy** - Jeden sąsiad powyżej, drugi poniżej.

Wielokąt jest y-monotoniczny wtedy i tylko wtedy gdy nie posiada żadnego wierzchołka łączącego, ani dzielącego.

Algorytm triangulacji opisany jest następująco:

- Przydzielamy każdy z wierzchołków do łańcucha
- Sortujemy wierzchołki względem współrzędnej y
- Umieszczamy dwa pierwsze wierzchołki na stosie

- Jeśli sprawdzany wierzchołek należy do innego łańucha od tego ze szczytu stosu, to łączymy go ze wszystkimi wierzchołkami na stosie, następnie wstawiamy na stos wierzchołek który był na szczycie oraz ten który był właśnie przetwarzany
- Jeśli sprawdzany wierzchołek należy do tego samego łańcucha, to analizujemy trójkąt jaki tworzy z wierzchołkami na szczycie stosu, jeśli zawiera się w tym wielokącie, to zdejmujemy jeden wierzchołek ze stosu i powtarzamy ten krok, w przeciwnym wypadku umieszczamy badane wierzchołki na stosie i przechodzimy do następnego wierzchołka

1.3 Opis implementacji

Sprawdzanie y-monotoniczności: Funkcja zakłada, że współrzędne wierzchołków są podane w postaci tablicy krotek i punkty są posortowane w kolejności przeciwnej do ruchu wskazówek zegara. Funkcja najpierw znajduje wierzchołek o najwyższym i najniższym y, następnie dzieli wielokąt na dwa łańcuchy zaczynające i kończące się w tych wierzchołkach, następnie sprawdza czy oba są monotoniczne, jeśli tak, to wielokąt jest y-monotoniczny.

Klasyfikacja wierzchołków: Funkcja iteruje przez wszystkie wierzchołki sprawdzając położenie sąsiadów oraz wypukłość kąta, przy sprawdzanym wierzchołku. Do sprawdzania wypukłości kąta, jest używana metoda wyznacznika macierzy 3x3.

Triangulacja: Funkcja realizuje algorytm opisany we wstępie teoretycznym. Po podzieleniu wielokąta na dwa łańcuchy, wierzchołki są sortowane przez scalanie. W ramach stosu użyto struktury deque z biblioteki queque. Do sprawdzania czy dany wierzchołek należy do wielokąta użyto metody obliczania wyznacznika macierzy 3x3. Przeanalizowano skuteczność algorytmu dla różnych wielokątów.

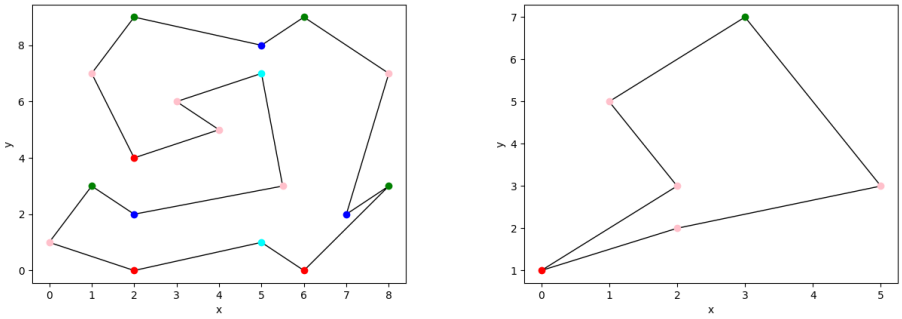
2 Wyniki

2.1 Klasyfikacja wierzchołków

Poniżej dwie grafiki pokazujące na przykładzie jak wygląda klasyfikacja wierzchołków w wielokącie:

Legenda:

- **Początkowy**
- **Końcowy**
- **Łączący**
- **Dzielący**
- **Prawidłowy**

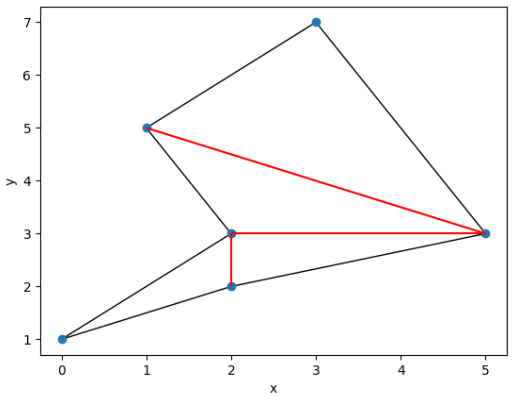


(a) Wielokąt dowolny (b) Wielokąt y-monotoniczny

Rysunek 1: Klasyfikacja wierzchołków

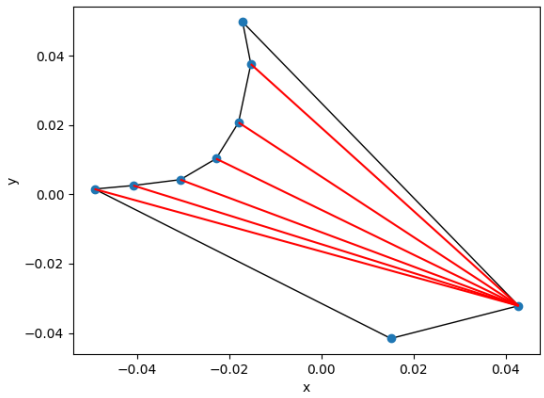
2.2 triangulacja

Algorytm triangulacji wielokątów y-monotonicznych został przetestowany na zróżnicowanym zestawie figur, aby zweryfikować jego poprawność. Testy wizualizują przekątne (zaznaczone kolorem czerwonym) dodane przez algorytm.



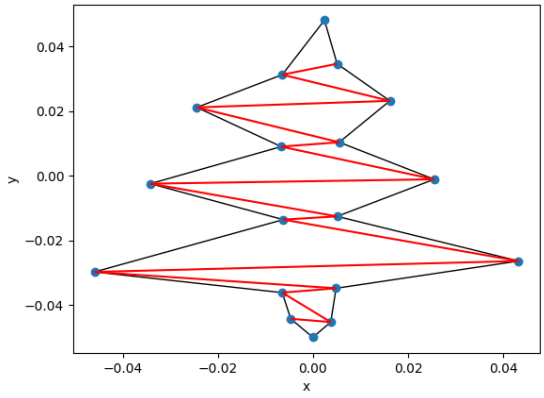
Rysunek 2: Wielokąt bazowy

Komentarz: Prosty, niesymetryczny wielokąt y-monotoniczny. Algorytm poprawnie połączył wierzchołki.



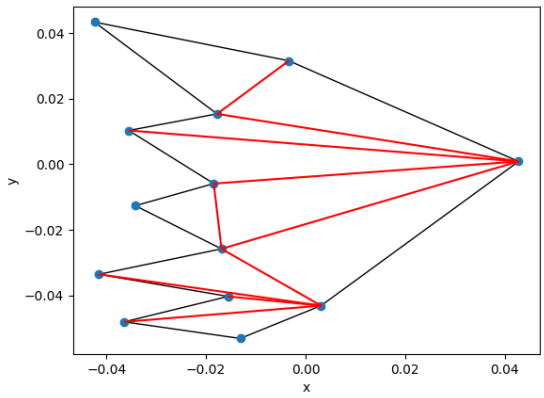
Rysunek 3: Wachlarz

Komentarz: Jest to najprostszy przypadek wklęsłości (jeden łańcuch jest prostą linią), w którym algorytm tworzy wachlarz trójkątów z jednym wspólnym wierzchołkiem. Triangulacja została poprawnie wykonana.



Rysunek 4: Choinka

Komentarz: Wielokąt z podobnymi wklęsłościami po obu stronach. Algorytm pomyślnie poradził sobie z zarządzaniem oboma łańcuchami, tworząc trójkąty w miarę wychodzenia w górę (wzrostu y).



Rysunek 5: Grzebień

Komentarz: Reprezentuje trudniejszy przypadek, w którym występuje wiele następujących po sobie wklęsłych narożników na jednym z łańcuchów. Również tutaj, algorytm sobie poradził

Wyniki Triangulacji dla Różnych Wielokątów Testowych

3 Wnioski

Implementacja algorytmu triangulacji wielokątów y-monotonicznych przebiegła pomyślnie. Algorytm triangulacji, bazujący na sortowaniu wierzchołków i wykorzystaniu stosu, efektywnie i optymalnie czasowo ($\mathbf{O(n)}$) generuje zbiór przekątnych, poprawnie dzieląc dostarczone wielokąty y-monotoniczne na trójkąty.

4 Środowisko

- System operacyjny: Linux 6.17.9-arch1-1
- Procesor: AMD Ryzen 9800x3d
- Język programowania + translator: Python 3.13.7
- Środowisko: JupyterNotebook
- wynik !jupyter –version
  - IPython : 9.7.0
  - ipykernel : 7.1.0
  - ipywidgets : not installed
  - jupyter_client : 8.6.3
  - jupyter_core : 5.9.1
  - jupyter_server : 2.17.0
  - jupyterlab : 4.5.0
  - nbclient : 0.10.2

- nbconvert : 7.16.6
- nbformat : 5.10.4
- notebook : 7.5.0
- qtconsole : not installed
- traitlets : 5.14.3