Wieloboki Voronoi

Porównanie metod konstrukcji

Jakub Pawlina, Bartosz Hanc

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

styczeń 2023

Plan

Opis problemu

Opis problemu

Diagramem Woronoja generowanym przez zbiór punktów $S := \{p_1, ..., p_n\} \subset \mathbb{R}^2$ nazywamy podział płaszczyzny $\mathbb{R}^2$ na obszary H_1,...,H_n parami rozłączne, takie że dla każdego H_i zachodzi $\forall_{p \in H_i} : d(p, p_i) = \min\{d(p, p_j) \,|\, p_j \in S\}$, gdzie $d : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}_+$ jest metryką na płaszczyźnie $\mathbb{R}^2$.

Algorytm Bowyera - Watsona

Opis algorytmu

Ideą metody konstrukcji diagramu Woronoja jest wykorzystanie dualności między triangulacją Delaunay'a, a diagramem Woronoja. Wyznaczamy najpierw triangulację Delaunay'a chmury punktów korzystając z algorytmu Bowyera–Watsona, a następnie w czasie liniowym przekształcamy uzyskaną triangulację w diagram Woronoja.

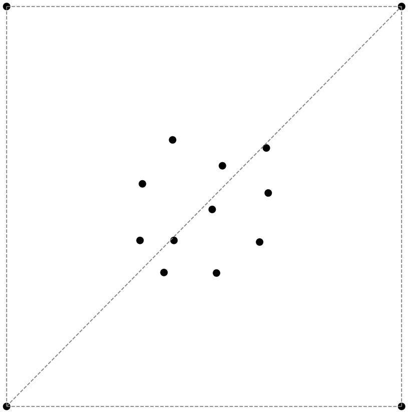
Opis algorytmu

```
funkcja BowyerWatson:
    points := lista zadanych punktów
    triangles := pusty zbiór trójkątów triangulacji

    Znajdź super-prostokąt pokrywający całkowicie chmurę punktów.
    Dodaj do triangles dwa super-trójkąty powstałe z podziału
    super-prostokąta jedną z jego przekątnych.

    Dla każdego punktu p z points:
        Znajdź wszystkie trójkąty T, takie że okrąg opisany na T zawiera
        punkt p i dodaj je do zbioru badTriangles.

        Znajdź krawędzie wielokąta, który powstałby po usunięciu trójkątów
        z badTriangles i dodaj je do zbioru polygon.

        Usuń z triangles wszystkie trójkąty znajdujące się w badTriangles.

        Dokonaj triangulacji powstałej wielokątnej wnęki polygon.

    Usuń z triangles wszystkie trójkąty, które zawierają wierzchołek
    super-trójkąta.

    Oblicz diagram Woronoja jako graf dualny do triangulacji Delaunaya
    zawartej w triangles.
```

Wykorzystane struktury danych

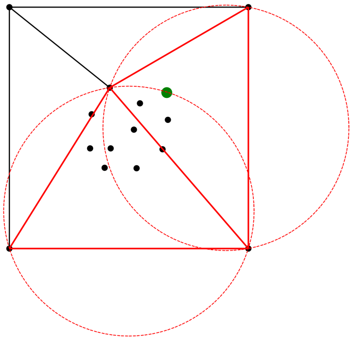
- Wejściowe punkty są przechowywane w liście `points`, która zawiera pary typu `(float,float)`.
- Struktura `triangles` przechowuje trójki postaci `(int,int,int)` określające wierzchołki trójkąta.
- Struktura `edges` przechowuje pary typu klucz-wartość, gdzie kluczami są pary typu `(int,int)` określające krawędź triangulacji jako indeksy punktów w liście `points`, natomiast wartościami są zbiory zawierające trójkąty, które zawierają krawędź będąca kluczem.

Konstrukcja super-prostokąta



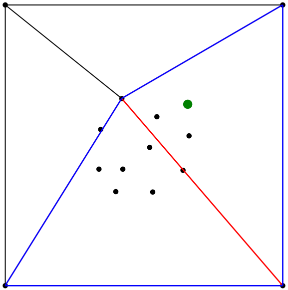
Poszukiwanie trójkątów, których okrąg opisany zawiera dany punkt p

- Dla każdego trójkąta $T(A, B, C)$ znajdującego się aktualnie w zbiorze `triangles` obliczamy współrzędne (S_x, S_y) środka okręgu opisanego na nim i jego promień R.
- Jeśli odległość punktu p od środka $\texttt{centre}(A, B, C)$ jest mniejsza lub równa R to dodajemy trójkąt $T(A, B, C)$ do zbioru `badTriangles`.



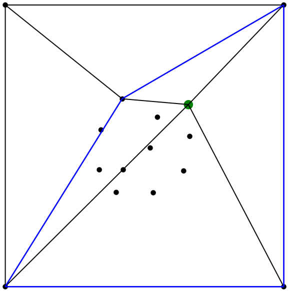
Poszukiwanie krawędzi wielokątnej wnęki

- Dla każdego trójkąta w `badTriangles` znajdujemy krawędź, która nie jest wspólna dla żadnych dwóch trójkątów w `badTriangles`. Krawędź ta jest krawędzią wielokątnej wnęki.

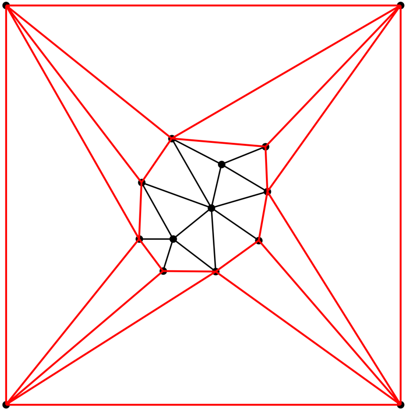


Usunięcie nieprawidłowych trójkątów i triangulacja wnęki

- Usuwamy z `triangles` wszystkie trójkąty znajdujące się w `badTriangles`.
- Dla każdej krawędzi $e = (q, r)$ ze zbioru wielokątnej wnęki dodajemy do zbioru `triangles` trójkąt $T(p, q, r)$ i aktualizujemy strukturę `edges`.

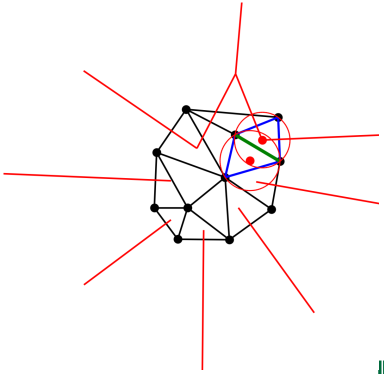


Usunięcie trójkątów zawierających wierzchołek super-prostokąta

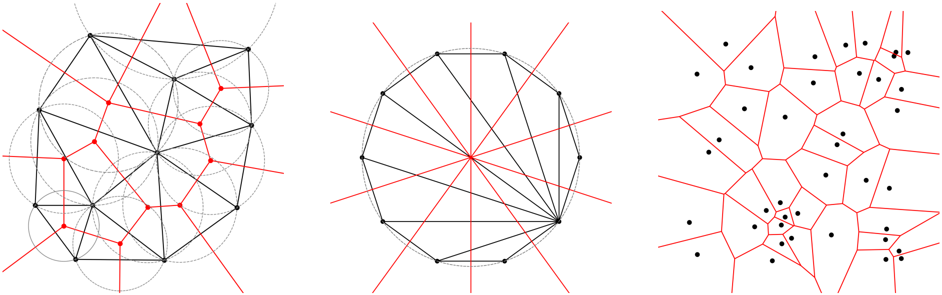


Transformacja triangulacji w diagram Woronoja w $O(n)$

- Dla każdej krawędzi e będącej kluczem w `edges`:
  - Jeśli $\#\texttt{edges}[e] = 2$ to krawędź e jest krawędzią wspólną pewnych dwóch trójkątów T_1, T_2 triangulacji zatem do listy `voronoi` dodajemy odcinek łączący środki okręgów opisanych na T_1 i T_2.
  - Jeśli $\#\texttt{edges}[e] = 1$ to krawędź e należy do otoczki wypukłej wprowadzonej chmury punktów, a krawędź diagramu Woronoja jest fragmentem prostej prostopadłej do e.



Wizualizacja wyników

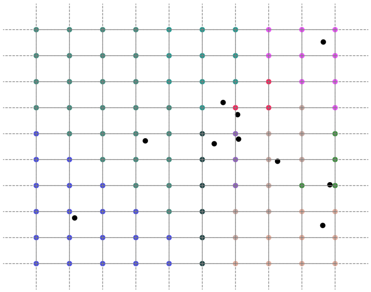


Rysunek: Wizualizacja wyznaczonego diagramu Woronoja dla trzech różnych zbiorów punktów

Algorytm przybliżony brute force

Opis algorytmu

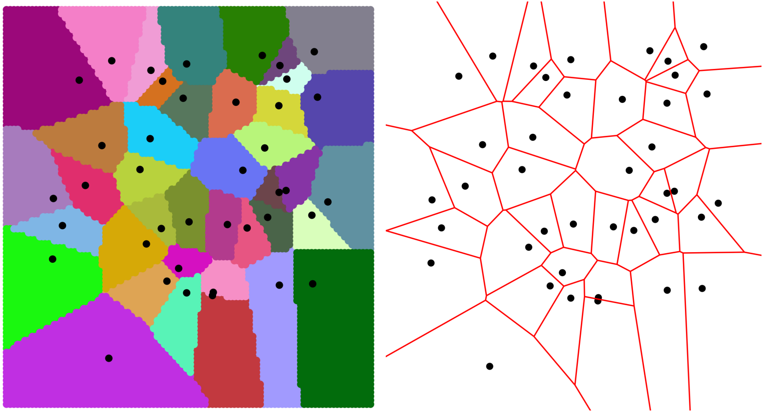
Algorytm wprowadza na płaszczyźnie dyskretną kratę $\mathbb{Z}_+^2$ i dla każdego punktu kratowego znajduje punkt z zadanej chmury, dla którego odległość (w sensie metryki euklidesowej) jest minimalna. Krata jest reprezentowana przez tablicę dwuwymiarową `grid`$[N][N]$, której wartościami są indeksy punktów z zadanej chmury `points`. Liczba naturalna N^2 określa liczbę punktów kratowych i jednocześnie wpływa bezpośrednio na dokładność wyznaczonego podziału.



Opis algorytmu

```python
def ApproxVoronoi(points, N): # O(n*N^2)
    n = len(points)
    P = [i for i in range(n)]
    mx, my = min(points, key=lambda x: x[0])[0], min(points, key=lambda x: x[1])[1]
    Mx, My = max(points, key=lambda x: x[0])[0], max(points, key=lambda x: x[1])[1]
    a = .15 * max(Mx-mx, My-my)
    mx -= a
    my -= a
    Mx += a
    My += a
    hx, hy = (Mx-mx)/N, (My-my)/N
    grid = [[min(P, key=lambda x: d(points[x], (mx+i*hx, my+j*hy))) for j in range(N)] for i in range(N)]
    return grid, mx, my, hx, hy
```

Wizualizacja wyników



Rysunek: Wizualizacja wyznaczonego przybliżonego diagramu Woronoja dla zbioru 40 losowych punktów oraz diagram dokładny dla tego zbioru punktów.

Źródła

1. Wikipedia. *Bowyer–Watson algorithm.* https://en.wikipedia.org/wiki/Bowyer-Watson_algorithm.
2. Wikipedia. *Voronoi diagram.* https://en.wikipedia.org/wiki/Voronoi_diagram.