

Geometria obliczeniowa

Krzysztof Gdawiec

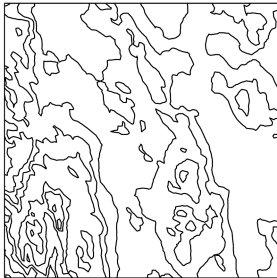
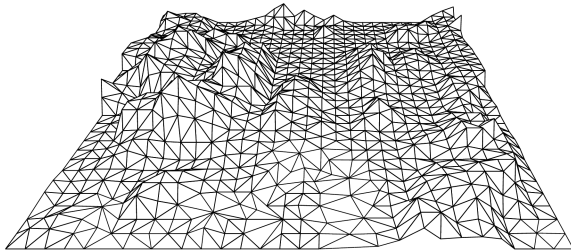


UNIWERSYTET ŚLĄSKI
INSTYTUT INFORMATYKI

Triangulacja Delaunay

Skupimy się na tworzeniu modelu fragmentu powierzchni ziemi czyli terenu. Teren jest powierzchnią dwuwymiarową w przestrzeni trójwymiarowej o tej własności, że każda pionowa prosta przecina go w co najwyżej jednym punkcie. Zatem jest to wykres funkcji $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, która wyznacza wysokość $f(p)$ dla każdego $p \in A$.

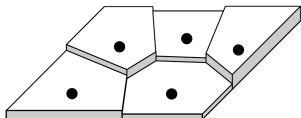
Teren możemy przedstawiać na różne sposoby, np. za pomocą rysunku perspektywicznego czy też za pomocą konturu linii łączących punkty o tej samej wysokości (izolinie) tak jak na mapie topograficznej.



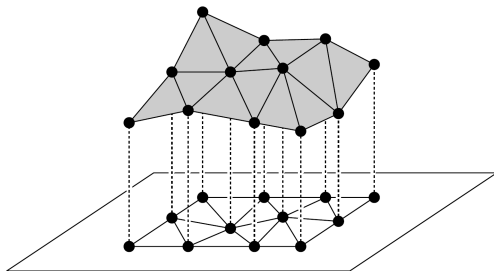
Oczywiście nie znamy wysokości każdego punktu na Ziemi, znamy ją tylko w punktach, w których ją zmierzaliśmy. Zatem mówiąc o pewnym terenie znamy wartości funkcji f na skończonym zbiorze $P \subset A$ punktów pomiarowych.

Na podstawie wysokości punktów pomiarowych musimy znaleźć aproksymację punktów w innych punktach dziedziny.

Naiwna metoda polega na przydzieleniu $p \in A$ wysokości najbliższego punktu pomiarowego. Nie będzie to wyglądać zbyt naturalnie.

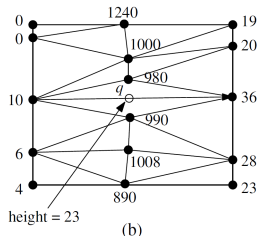
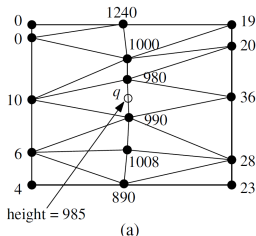


Najpierw określamy **triangulację** P czyli podział planarny, którego ograniczone ściany są trójkątami, a wierzchołki są punktami z P .
Potem podnosimy każdy punkt pomiarowy na jego właściwą wysokość.



W ten sposób otrzymujemy **teren wielościenny** (ang. polyhedral terrain), tj. wykres funkcji, która jest kawałkami liniowa.

Triangulacji punktów pomiarowych możemy dokonać na wiele różnych sposobów. Ponieważ nie znamy rzeczywistego terenu, więc nie jesteśmy w stanie stwierdzić, która jest najlepsza. Niemniej pewne triangulacje wyglądają bardziej naturalnie od innych.



Triangulacja (a) jest bardziej naturalna ponieważ modeluje górski grzbiet, a (b) wprowadza wąską dolinę przecinającą górski grzbiet.

Problem z triangulacją (b) jest taki, że wysokość q określana jest przez dwa punkty, które znajdują się względnie daleko. Punkt q leży po środku dwóch długich i ostrych trójkątów.

Triangulacje zawierające małe kąty są złe. Dlatego będziemy szeregować triangulacje przez porównywanie najmniejszych kątów. Jeśli najmniejsze kąty są takie same to patrzymy na drugi najmniejszy kąt itp.

Liczba triangulacji jest skończona, więc musi istnieć optymalna triangulacja maksymalizująca minimalny kąt. Jest to triangulacja, której szukamy.

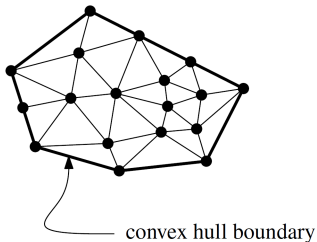
Triangulacje planarnych zbiorów punktów

Niech $P = \{p_1, p_2, \dots, p_n\}$ będzie zbiorem punktów na płaszczyźnie.

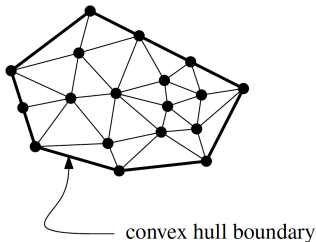
Maksymalnym podziałem planarnym (ang. maximal planar subdivision) S nazywamy podział taki, że nie można dodać do S żadnej krawędzi łączącej dwa wierzchołki bez zniszczenia jego planarności.

Triangulacją zbioru punktów P nazywamy maksymalny podział planarny, którego zbiorem wierzchołków jest P .

Dowolny odcinek łączący dwa kolejne punkty na brzegu otoczki wypukłej zbioru P jest krawędzią dowolnej triangulacji \mathcal{T} . To oznacza, że suma ograniczonych ścian w \mathcal{T} jest zawsze otoczką wypukłą zbioru P .



Dowolny odcinek łączący dwa kolejne punkty na brzegu otoczki wypukłej zbioru P jest krawędzią dowolnej triangulacji \mathcal{T} . To oznacza, że suma ograniczonych ścian w \mathcal{T} jest zawsze otoczką wypukłą zbioru P .



Liczba trójkątów i krawędzi w triangulacji jest stała i wynosi $2n - 2 - k$ trójkątów i $3n - 3 - k$ krawędzi, gdzie $n = \text{card}(P)$, a k jest to liczba punktów z P leżących na brzegu $\mathcal{CH}(P)$.

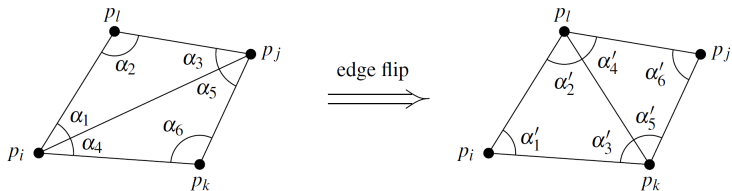
Niech \mathcal{T} będzie triangulacją P i założmy, że ma ona m trójkątów. W triangulacji \mathcal{T} mamy zatem $3m$ kątów. Niech $\alpha_1, \alpha_2, \dots, \alpha_{3m}$ będzie niemalejącym ciągiem tych kątów.

Niech $A(\mathcal{T}) = (\alpha_1, \alpha_2, \dots, \alpha_{3m})$ oznacza **wektor kątów** \mathcal{T} .

Założmy, że mamy drugą triangulację \mathcal{T}' z wektorem kątów $A(\mathcal{T}') = (\alpha'_1, \alpha'_2, \dots, \alpha'_{3m})$. Mówimy, że wektor kątów \mathcal{T} jest większy od wektora kątów \mathcal{T}' jeśli $A(\mathcal{T})$ jest leksykograficznie większe niż $A(\mathcal{T}')$. Oznaczamy to jako $A(\mathcal{T}) > A(\mathcal{T}')$.

Mówimy, że triangulacja \mathcal{T} jest **kątowo optymalna** (ang. angle-optimal), gdy $A(\mathcal{T}) \geq A(\mathcal{T}')$ dla wszystkich triangulacji \mathcal{T}' zbioru P .

Rozważmy krawędź $e = \overline{p_i p_j}$ triangulacji \mathcal{T} zbioru P . Jeśli e nie jest krawędzią nieograniczonej ściany, to sąsiaduje z dwoma trójkątami $p_i p_j p_k$ i $p_i p_j p_l$. Jeśli te dwa trójkąty tworzą czworokąt wypukły, to możemy otrzymać nową triangulację \mathcal{T}' przez usunięcie $\overline{p_i p_j}$ z \mathcal{T} i wstawienie $\overline{p_k p_l}$. Operację tę nazywamy **przekręcaniem krawędzi** (ang. edge flip).



Jedyną różnicę między $A(\mathcal{T})$ i $A(\mathcal{T}')$ stanowi sześć kątów $\alpha_1, \dots, \alpha_6$ w $A(\mathcal{T})$, które są zastępowane przez $\alpha'_1, \dots, \alpha'_6$ w $A(\mathcal{T}')$.

Krawędź $e = \overline{p_i p_j}$ nazywamy **krawędzią nielegalną** (ang. illegal edge), jeśli

$$\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i.$$

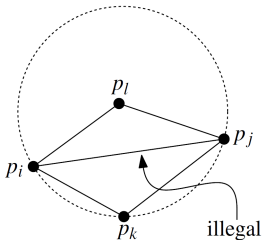
Czyli krawędź jest nielegalna, gdy przez przekroczenie tej krawędzi możemy lokalnie zwiększyć najmniejszy kąt.

Spostrzeżenie. Niech \mathcal{T} będzie triangulacją z krawędzią nielegalną e . Niech \mathcal{T}' będzie triangulacją otrzymaną z \mathcal{T} przez przekroczenie e . Wówczas

$$A(\mathcal{T}') > A(\mathcal{T}).$$

Okazuje się, że wcale nie musimy obliczać kątów, aby sprawdzić czy dana krawędź jest legalna. Pomocny w tym jest następujący lemat.

Lemat. Niech krawędź $e = \overline{p_i p_j}$ leży między trójkątami $p_i p_j p_k$ i $p_i p_j p_l$ oraz niech C będzie okręgiem przechodzącym przez p_i , p_j i p_k . Krawędź e jest nielegalna \iff punkt p_l leży wewnątrz C . Ponadto, jeśli punkty p_i , p_j , p_k , p_l tworzą czworokąt wypukły i nie leżą na tym samym okręgu, to dokładnie jedna z krawędzi $\overline{p_i p_j}$ oraz $\overline{p_k p_l}$ jest nielegalna.



Gdy wszystkie cztery punkty leżą na okręgu to obie krawędzie $\overline{p_i p_j}$ i $\overline{p_k p_l}$ są legalne.

Dwa trójkąty sąsiadujące z nielegalną krawędzią muszą tworzyć czworokąt wypukły, więc zawsze jest możliwe przekroczenie krawędzi nielegalnej.

Triangulację nazywamy **legalną** jeśli nie zawiera krawędzi nielegalnych. Zatem każda triangulacja kątowno optymalna jest legalna.

Mając daną triangulację możemy z niej otrzymać triangulację legalną przekraczając jej krawędzie nielegalne, aż wszystkie krawędzie staną się legalne.

Algorithm LEGALTRIANGULATION(\mathcal{T})

Input. Some triangulation \mathcal{T} of a point set P .

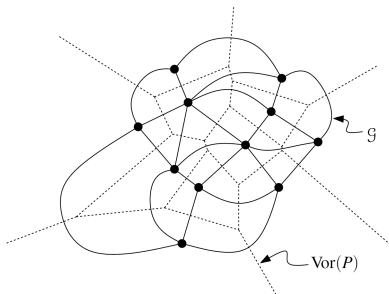
Output. A legal triangulation of P .

1. **while** \mathcal{T} contains an illegal edge $\overline{p_i p_j}$
2. **do** (* Flip $\overline{p_i p_j}$ *)
3. Let $p_i p_j p_k$ and $p_i p_j p_l$ be the two triangles adjacent to $\overline{p_i p_j}$.
4. Remove $\overline{p_i p_j}$ from \mathcal{T} , and add $\overline{p_k p_l}$ instead.
5. **return** \mathcal{T}

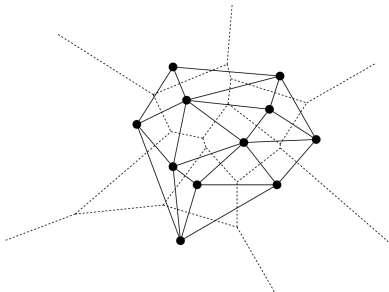
Triangulacja Delaunay

Niech P będzie zbiorem n punktów na płaszczyźnie. Przyjrzymy się teraz grafowi dualnemu do diagramu Voronoi zbioru P .

Graf dualny \mathcal{G} ma wierzchołki odpowiadające każdej komórce Voronoi i krawędzie między dwoma wierzchołkami, gdy odpowiednie komórki rozdziela krawędź.



Rozważmy prostoliniowe zanurzenie grafu \mathcal{G} , w którym wierzchołek odpowiadający komórce Voronoi $V(p)$ jest punktem p , a krawędź łącząca wierzchołki odpowiadające $V(p)$ i $V(q)$ jest odcinkiem \overline{pq} .



Zanurzenie to nazywamy **grafem Delaunay** zbioru P i oznaczamy $\mathcal{DG}(P)$.

Twierdzenie. Graf Delaunay zbioru punktów na płaszczyźnie jest grafem planarnym.

Jeśli wierzchołek Voronoi v jest wierzchołkiem komórek Voronoi dla centrów p_1, p_2, \dots, p_k , to odpowiednia ściana f w $\mathcal{DG}(P)$ ma p_1, p_2, \dots, p_k jako wierzchołki. Ponadto punkty p_1, p_2, \dots, p_k leżą na okręgu wokół v , więc f jest wielokątem wypukłym.

Jeśli punkty z P są rozmieszczone losowo szansa, że cztery punkty będą leżeć na jednym okręgu jest bardzo mała. Będziemy mówić, że zbiór punktów jest w **położeniu ogólnym** (ang. general position) jeśli nie zawiera czterech punktów współokręgowych.

Jeśli P jest w położeniu ogólnym, to wszystkie wierzchołki diagramu Voronoi mają stopień trzy. Zatem wszystkie ściany ograniczone $\mathcal{DG}(P)$ są trójkątami.

Triangulacją Delaunay nazywamy triangulację otrzymaną przez dodanie krawędzi do grafu Delaunay.

Triangulacja Delaunay jest jednoznaczna $\iff \mathcal{DG}(P)$ jest triangulacją, co ma miejsce, gdy P jest w położeniu ogólnym.

Twierdzenie. Niech P będzie zbiorem punktów na płaszczyźnie.

- ▶ Trzy punkty $p_i, p_j, p_k \in P$ są wierzchołkami tej samej ściany grafu Delaunay zbioru $P \iff$ okrąg przechodzący przez p_i, p_j, p_k nie zawiera w swoim wnętrzu żadnego punktu z P .
- ▶ Dwa punkty $p_i, p_j \in P$ tworzą krawędź grafu Delaunay zbioru $P \iff$ istnieje koło domknięte C , które zawiera p_i i p_j na swoim brzegu oraz nie zawiera żadnego innego punktu z P .

Z tego twierdzenia wynika następujące twierdzenie.

Twierdzenie. Niech P będzie zbiorem punktów na płaszczyźnie, a \mathcal{T} triangulacją zbioru P . \mathcal{T} jest triangulacją Delaunay zbioru $P \iff$ okrąg opisany na dowolnym trójkącie \mathcal{T} nie zawiera w swoim wnętrzu punktu z P .

Twierdzenie. Niech P będzie zbiorem punktów na płaszczyźnie. Triangulacja \mathcal{T} zbioru P jest legalna $\iff \mathcal{T}$ jest triangulacją Delaunay zbioru P .

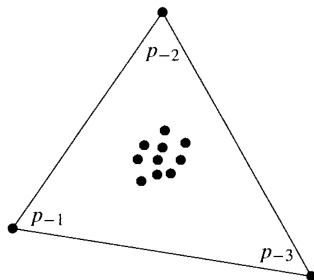
Ponieważ dowolna triangulacja kątowno optymalna musi być legalna, więc zgodnie z powyższym twierdzeniem musi być triangulacją Delaunay.

Gdy P jest w położeniu ogólnym istnieje tylko jedna triangulacja legalna, która jest wtedy jedyną triangulacją kątowno optymalną. Gdy P nie jest w położeniu ogólnym, to dowolna triangulacja Delaunay jest legalna.

Obliczanie triangulacji Delaunay

Algorytm obliczania triangulacji Delaunay, który za chwilę wprowadzimy będzie randomizowanym algorytmem przyrostowym.

Pierwsze co robimy to tworzymy trójkąt o wierzchołkach p_{-1} , p_{-2} , p_{-3} zawierający zbiór P .



Będziemy obliczać triangulację Delaunay zbioru $P \cup \{p_{-1}, p_{-2}, p_{-3}\}$. Po jej wyznaczeniu odrzucamy wierzchołki p_{-1}, p_{-2}, p_{-3} wraz ze wszystkimi krawędziami incydentnymi otrzymując triangulację Delaunay zbioru P .

W algorytmie będziemy dodawać punkty w porządku losowym i będziemy pamiętać triangulację Delaunay dla aktualnego zbioru punktów.

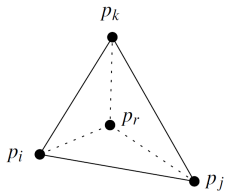
W jaki sposób dodajemy punkty?

Założmy, że chcemy dodać punkt p_r . Najpierw znajdujemy trójkąt w aktualnej triangulacji, który zawiera p_r .

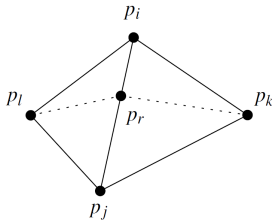
Mogą zajść dwa przypadki:

- ▶ punkt należy do wnętrza trójkąta – dodajemy krawędzie z p_r do wierzchołków trójkąta,
- ▶ punkt leży na krawędzi e trójkąta – dodajemy krawędzie z p_r do przeciwległych wierzchołków trójkątów zawierających e .

p_r lies in the interior of a triangle



p_r falls on an edge



Tak powstała triangulacja niekoniecznie musi być triangulacją Delaunay, ponieważ dodanie p_r mogło uczynić niektóre istniejące krawędzie nielegalnymi.

Aby poprawić triangulację będziemy musieli poprawić potencjalne nielegalne krawędzie. Zrobimy to za pomocą funkcji `LEGALIZEEDGE`.

Procedura ta zastępuje krawędzie nielegalne przez legalne za pomocą przekręceń krawędzi.

Tak powstała triangulacja niekoniecznie musi być triangulacją Delaunay, ponieważ dodanie p_r mogło uczynić niektóre istniejące krawędzie nielegalnymi.

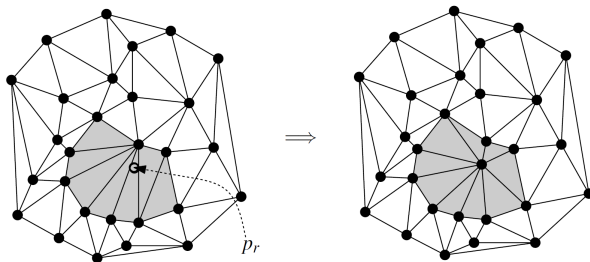
Aby poprawić triangulację będziemy musieli poprawić potencjalne nielegalne krawędzie. Zrobimy to za pomocą funkcji `LEGALIZEEDGE`.

Procedura ta zastępuje krawędzie nielegalne przez legalne za pomocą przekręceń krawędzi.

Które krawędzie mogą stać się nielegalne po wstawieniu p_r ?

Zauważmy, że krawędź $\overline{p_i p_j}$, która przedtem była legalna może stać się nielegalna tylko gdy jeden z trójkątów sąsiadujących z nią został zmieniony.

Zatem musimy sprawdzić tylko krawędzie nowych trójkątów. Wykonujemy to używając funkcji `LEGALIZEEDGE`, która sprawdza i być może przekręca krawędź. Jeśli przekręcimy krawędź, to inne krawędzie mogą stać się nielegalne. Dlatego wywołujemy rekurencyjnie `LEGALIZEEDGE` dla potencjalnie nielegalnych krawędzi.



Algorytm 1: Triangulacja Delaunay

Dane: Zbiór P zawierający $n \in \mathbb{N}$ punktów przestrzeni \mathbb{R}^2 .

Wynik: Triangulacja Delaunay T zbioru P .

- 1 Niech (x_l, y_l) , (x_u, y_u) będą lewym dolnym i prawym górnym rogiem prostokąta zawierającego punkty z P
 - 2 $(x_b, y_b) = ((x_l + x_u)/2, (y_l + y_u)/2)$
 - 3 $d = 100 \cdot \max\{|x_u - x_l|, |y_u - y_l|\}$
 - 4 $p_{-1} = (x_b - d, y_b - d)$, $p_{-2} = (x_b, y_b + d)$, $p_{-3} = (x_b + d, y_b - d)$
 - 5 Inicjuj T jako triangulację zawierającą pojedynczy trójkąt o wierzchołkach p_{-1}, p_{-2}, p_{-3}
 - 6 Oblicz losową permutację punktów $p_1, p_2, \dots, p_n \in P$
 - 7 **for** $r = 1$ **to** n **do**
 - 8 Znajdź trójkąt $T \in \mathcal{T}$ zawierający p_r ; oznaczmy jego wierzchołki przez p_i, p_j, p_k
 - 9 **if** p_r *leży we wnętrzu trójkąta* T **then**
 - 10 Dodaj krawędzie z p_r do trzech wierzchołków p_i, p_j, p_k dzieląc w ten sposób T na trzy trójkąty
 - 11 LEGALIZEEDGE($p_r, \overline{p_i p_j}$, T)
 - 12 LEGALIZEEDGE($p_r, \overline{p_j p_k}$, T)
 - 13 LEGALIZEEDGE($p_r, \overline{p_k p_i}$, T)
 - 14 **else**
 - 15 Dodaj krawędzie z p_r do p_k i do trzeciego wierzchołka p_l drugiego trójkąta sąsiadującego z krawędzią $\overline{p_i p_j}$, dzieląc w ten sposób dwa trójkąty sąsiadujące z $\overline{p_i p_j}$ na cztery trójkąty
 - 16 LEGALIZEEDGE($p_r, \overline{p_i p_l}$, T)
 - 17 LEGALIZEEDGE($p_r, \overline{p_l p_j}$, T)
 - 18 LEGALIZEEDGE($p_r, \overline{p_j p_k}$, T)
 - 19 LEGALIZEEDGE($p_r, \overline{p_k p_i}$, T)
 - 20 Usuń z T punkty p_{-1}, p_{-2}, p_{-3} wraz ze wszystkimi incydentnymi z nimi krawędziami
 - 21 **return** T
-

Algorytm 3: LEGALIZEEDGE

Dane: p_r jest wstawianym punktem, \mathcal{T} jest triangulacją, $\overline{p_i p_j}$ jest krawędzią w \mathcal{T} , która może wymagać przekroczenia.

```
1 LEGALIZEEDGE( $p_r, \overline{p_i p_j}, \mathcal{T}$ )
2   if  $\overline{p_i p_j}$  jest nielegalna then
3     Niech  $T$  będzie trójkątem o wierzchołkach  $p_i, p_j, p_k$  sąsiadującym z
4     trójkątem  $T'$  o wierzchołkach  $p_r, p_i, p_j$  wzdłuż krawędzi  $\overline{p_i p_j}$ 
5     Zastąp  $\overline{p_i p_j}$  przez  $\overline{p_r p_k}$ .
6     LEGALIZEEDGE( $p_r, \overline{p_i p_k}, \mathcal{T}$ )
7     LEGALIZEEDGE( $p_r, \overline{p_k p_j}, \mathcal{T}$ )
```

Algorytm 4: Sprawdzenie czy krawędź jest nielegalna

Dane: p_i, p_j końce krawędzi.

Wynik: FALSE – krawędź jest nielegalna, TRUE – krawędź jest legalna.

```
1 if  $i, j < 0$  then
2   return TRUE
3 Niech  $p_k, p_l$  będą pozostałymi wierzchołkami trójkątów współdzielących krawędź
4    $\overline{p_i p_j}$ 
5   if  $p_l$  leży wewnątrz okręgu opisanego na trójkącie o wierzchołkach  $p_i, p_j, p_k$  then
6     return FALSE
7   else
8     return TRUE
```

Pozostaje nam kwestia wyszukiwania trójkąta zawierającego wstawiany punkt p_r . Do tego celu używamy specjalnej struktury danych.

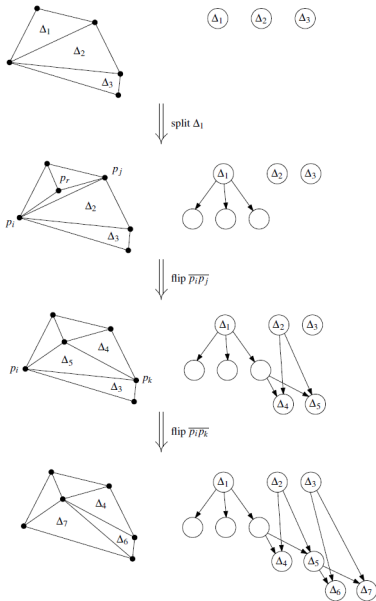
Podczas budowy triangulacji Delaunay tworzymy również strukturę \mathcal{D} lokalizacji punktów, która jest acyklicznym grafem skierowanym:

- ▶ liście odpowiadają trójkątom aktualnej triangulacji,
- ▶ węzły wewnętrzne odpowiada trójkątom, które były w triangulacji w pewnej wcześniejszej fazie, ale zostały już zmienione.

Inicjujemy \mathcal{D} jako acykliczny graf skierowany z jednym liściem, który odpowiada trójkątowi o wierzchołkach p_{-1}, p_{-2}, p_{-3} .

Założmy, że w pewnym punkcie dzielimy trójkąt Δ o wierzchołkach p_i, p_j, p_k aktualnej triangulacji na trzy (lub dwa) nowe trójkąty. W \mathcal{D} dodajemy trzy (lub dwa) nowe liście i zamieniamy liść odpowiadający Δ na węzeł wewnętrzny z wychodzącymi wskaźnikami do tych trzech (lub dwóch liści).

Przekręcając krawędź zastępujemy dwa trójkąty Δ_1, Δ_2 o wierzchołkach p_k, p_i, p_j i p_i, p_j, p_l (odpowiednio) przez trójkąty o wierzchołkach p_k, p_i, p_l i p_k, p_l, p_j , tworzymy liście dla dwóch nowych trójkątów, a węzły odpowiadające Δ_1 i Δ_2 otrzymują wskaźniki do tych dwóch nowych liści.



Mając graf \mathcal{D} możemy zlokalizować trójkąt zawierający p_r .
Zaczynamy w korzeniu struktury \mathcal{D} . Sprawdzamy trzech synów korzenia, aby zobaczyć, w którym trójkącie leży punkt p_r i schodzimy do odpowiedniego syna.

Następnie sprawdzamy synów tego węzła. Schodzimy do syna, którego trójkąt zawiera p_r itd. aż dotrzemy do liścia struktury \mathcal{D} . Liść ten odpowiada trójkątowi w aktualnej triangulacji, który zawiera punkt p_r .

Mając graf \mathcal{D} możemy zlokalizować trójkąt zawierający p_r . Zaczynamy w korzeniu struktury \mathcal{D} . Sprawdzamy trzech synów korzenia, aby zobaczyć, w którym trójkącie leży punkt p_r i schodzimy do odpowiedniego syna.

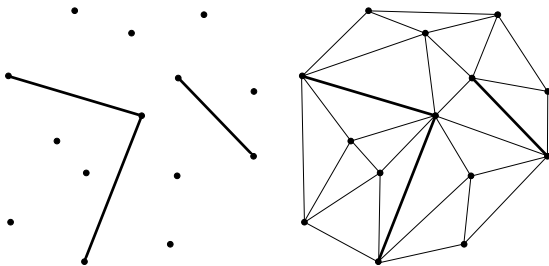
Następnie sprawdzamy synów tego węzła. Schodzimy do syna, którego trójkąt zawiera p_r itd. aż dotrzemy do liścia struktury \mathcal{D} . Liść ten odpowiada trójkątowi w aktualnej triangulacji, który zawiera punkt p_r .

Triangulację Delaunay zbioru P zawierającego n punktów na płaszczyźnie można obliczyć w oczekiwanym czasie $\mathcal{O}(n \log n)$ używając $\mathcal{O}(n)$ oczekiwanej pamięci.

Triangulacja Delaunay z ograniczeniami

Rozpatrzmy teraz następujący problem:

Mając dany zbiór punktów i zbiór nieprzecinających się krawędzi pomiędzy tymi punktami należy znaleźć triangulację Delaunay, która zawiera ten zbiór krawędzi.



Problem ten znany jest jako triangulacja Delaunay z ograniczeniami (ang. constrained Delaunay triangulation).

Niech $P = \{p_1, p_2, \dots, p_n\}$ będzie zbiorem punktów na płaszczyźnie oraz niech E_c będzie zbiorem krawędzi pomiędzy punktami z P , tzn. krawędź \overline{ab} należy do E_c jeśli $a, b \in P$.

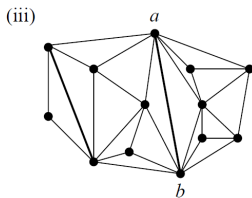
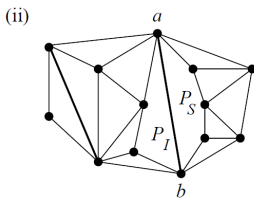
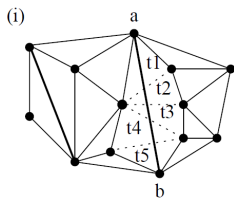
Szukamy triangulacji Delaunay, w której E_c jest podzbiorem krawędzi tej triangulacji.

Zastosujemy podejście przyrostowe, tzn.

- ▶ znajdziemy triangulację Delaunay T zbioru punktów P używając dowolnego algorytmu,
- ▶ wstawimy po kolei krawędzie z E_c do T dokonując odpowiednich zmian w triangulacji.

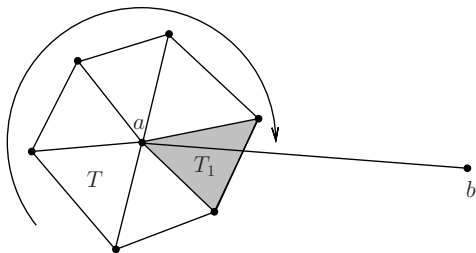
Wstawianie krawędzi $\overline{ab} \in E_c$ do triangulacji możemy przeprowadzić w następujących krokach:

- (i) z triangulacji usuwamy trójkąty t_1, t_2, \dots, t_k , które są przecięte przez krawędź \overline{ab} . Otrzymujemy obszar, który nie jest striangulowany;
- (ii) dodajemy krawędź \overline{ab} do triangulacji;
- (iii) dokonujemy triangulacji niestriangulowanego obszaru powyżej i poniżej \overline{ab} .



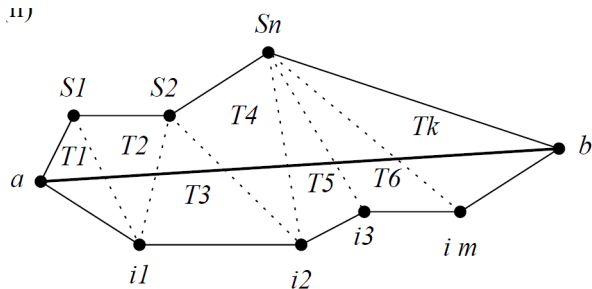
W pierwszym kroku musimy znaleźć wszystkie trójkąty t_1, t_2, \dots, t_k , które są przecięte przez krawędź \overline{ab} .

W tym celu najpierw znajdujemy dowolny trójkąt t w triangulacji, którego wierzchołkiem jest a . Następnie poruszając się zgodnie z ruchem wskazówek zegara (wykorzystujemy informacje o sąsiedztwie trójkątów) sprawdzamy inne trójkąty współdzielące a , aż nie znajdziemy trójkąta t_1 przeciętego przez krawędź \overline{ab} .



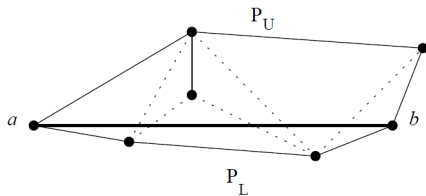
Ponownie wykorzystując informacje o sąsiedztwie znajdujemy pozostałe trójkąty przecięte przez \overline{ab} .

Podczas znajdowania trójkątów t_1, t_2, \dots, t_k przeciętych przez \overline{ab} możemy podzielić ich wierzchołki na dwa zbiory – poniżej $\{i_1, i_2, \dots, i_m\}$ i powyżej $\{s_1, s_2, \dots, s_n\}$ krawędzi \overline{ab} .



Zbiory wierzchołków $P_U = \{a, s_1, \dots, s_n, b\}$ i $P_L = \{a, i_1, \dots, i_m, b\}$ tworzą tzw. pseudo-wielokąty, które należy poddać triangulacji.

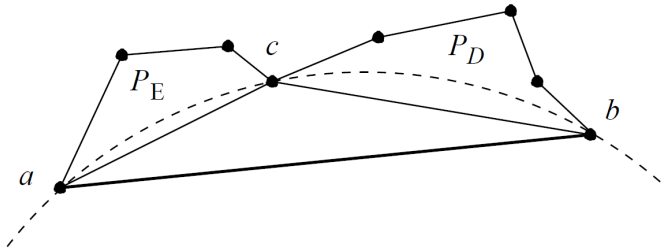
P_U i P_L nazywamy pseudo-wielokątami ponieważ mogą one zawierać powtórzone wierzchołki, np.



Zauważmy, że triangulując pseudo-wielokąt P_* musi powstać trójkąt, którego krawędzią jest \overline{ab} . Ponadto, triangulacja musi być triangulacją Delaunay, więc trójkąt ten musi spełniać warunek, który podaliśmy przy triangulacji Delnaunay, tzn. okrąg opisany na tym trójkącie nie może zawierać innych wierzchołków z P_* .

Korzystając z tej obserwacji dostajemy następujący algorytm rekurencyjny:

- ▶ znajdź wierzchołek $c = s_l$ taki, że okrąg opisany na trójkącie o wierzchołkach a, b, c nie zawiera żadnego innego punktu z P_* ,
- ▶ utwórz trójkąt o wierzchołkach a, b, c , który dzieli P_* na dwa mniejsze obszary $P_E = \{a, s_1, s_2, \dots, s_l\}$ i $P_D = \{s_l, s_{l+1}, \dots, s_n, b\}$,
- ▶ powtarzamy całość rekurencyjnie dla P_E i P_D względem krawędzi \overline{ac} i \overline{cb} , odpowiednio.



Algorytm 7: TRIANGULATEPSEUDOPOLYGONDELAUNAY

Dane: Zbiór (lista) wierzchołków P , krawędź \overline{ab} , triangulacja \mathcal{T} .

Wynik: Triangulacja wierzchołków ze zbioru P .

```
1 TRIANGULATEPSEUDOPOLYGONDELAUNAY( $P, \overline{ab}, \mathcal{T}$ )
2   | if  $|P| > 1$  then
3   |   |  $c =$  pierwszy wierzchołek z  $P$ 
4   |   | for każdego wierzchołka  $v \in P$  do
5   |   |   | if  $v$  leży wewnątrz okręgu opisanego na trójkącie o wierzchołkach  $a, b, c$ 
6   |   |   |   | then
7   |   |   |   |   |  $c = v$ 
8   |   |   |   |   |
9   |   |   |   |   | Podziel  $P$  na zbiory  $P_E$  i  $P_D$  takie, że  $P = P_E \cup \{c\} \cup P_D$ 
10  |   |   |   |   | TRIANGULATEPSEUDOPOLYGONDELAUNAY( $P_E, \overline{ac}, \mathcal{T}$ )
11  |   |   |   |   | TRIANGULATEPSEUDOPOLYGONDELAUNAY( $P_D, \overline{cb}, \mathcal{T}$ )
12  |   |   |   |   |
13  |   |   |   |   | if  $P \neq \emptyset$  then
14  |   |   |   |   |   | Dodaj trójkąt o wierzchołkach  $a, b, c$  do  $\mathcal{T}$ 
```

Niech

- ▶ `OPPOSEDTRIANGLE` będzie funkcją przyjmującą jako argumenty trójkąt T i jeden z jego wierzchołków v , a zwracającą trójkąt przyległy do T , który nie zawiera wierzchołka v .
- ▶ `OPPOSEDVERTEX` będzie funkcją przyjmującą jako argumenty dwa trójkąty T_s , T i zwracającą wierzchołek trójkąta T_s , który nie należy do T .

Algorytm 6: Wstawianie krawędzi ze zbioru E_c do triangulacji

Dane: Triangulacja \mathcal{T} zbioru punktów P , krawędź $\overline{ab} \in E_c$ taka, że $a, b \in P$ i $\overline{ab} \notin \mathcal{T}$.

Wynik: Triangulacja ze wstawioną krawędzią \overline{ab} .

- 1 Znajdź trójkąt $T \in \mathcal{T}$, który zawiera punkt a oraz jest przecięty przez krawędź \overline{ab}
 - 2 $P_U = P_L =$ pusta lista
 - 3 $v = a$
 - 4 **while** $b \notin T$ **do**
 - 5 $T_s = \text{OPPOSEDTRIANGLE}(T, v)$
 - 6 $v_s = \text{OPPOSEDVERTEX}(T_s, T)$
 - 7 **if** v_s *jest ponad krawędzią* \overline{ab} **then**
 - 8 Dodaj wierzchołek v_s do listy P_U
 - 9 $v =$ wierzchołek współdzielony przez T i T_s ponad krawędzią \overline{ab}
 - 10 **else**
 - 11 Dodaj wierzchołek v_s do listy P_L
 - 12 $v =$ wierzchołek współdzielony przez T i T_s poniżej krawędzi \overline{ab}
 - 13 Usuń trójkąt T z \mathcal{T}
 - 14 $T = T_s$
 - 15 TRIANGULATEPSEUDOPOLYGONDELAUNAY($P_U, \overline{ab}, \mathcal{T}$)
 - 16 TRIANGULATEPSEUDOPOLYGONDELAUNAY($P_L, \overline{ab}, \mathcal{T}$)
 - 17 Zrekonstruuuj sąsiedztwo trójkątów w \mathcal{T}
 - 18 Dodaj krawędź \overline{ab} do \mathcal{T}
 - 19 Oznacz krawędź \overline{ab} z \mathcal{T} jako stałą
-

Niech n będzie liczbą trójkątów w triangulacji, a e będzie liczbą trójkątów przeciętych przez krawędź \overline{ab} .

Pesymistyczny czas kroku znajdującego trójkąt triangulacji zawierający wierzchołek a i przecięty przez \overline{ab} wynosi $\mathcal{O}(n)$.

Pesymistyczny czas konstrukcji P_U i P_L wynosi $\mathcal{O}(e)$.

Pesymistyczny czas triangulacji pseudo-wielokąta wynosi $\mathcal{O}(e^2)$.