

Rovinné triangulace a jejich využití.

Greedy Triangulation. Delaunay Triangulation. Constrained Delaunay Triangulation. Data Dependent Triangulation. DMT.

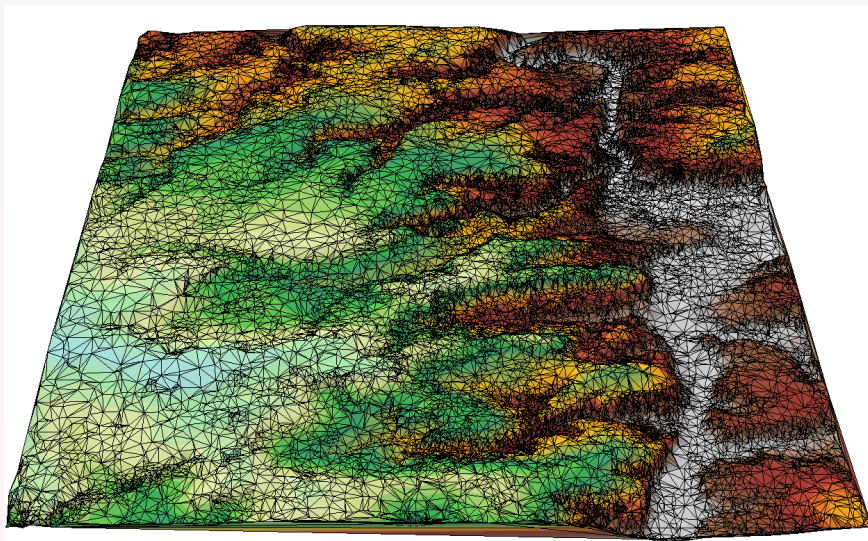
Tomáš Bayer | bayertom@natur.cuni.cz

Katedra aplikované geoinformatiky a kartografie. Přírodovědecká fakulta UK.

Obsah přednášky

- 1 Ukázka použití
- 2 Formulace problému
- 3 Vlastnosti triangulací
- 4 Greedy triangulace
- 5 Delaunay triangulace
 - Metoda inkrementální konstrukce
 - Metoda inkrementálního vkládání
- 6 Triangulace se vstupní podmínkou
- 7 Datově závislé triangulace
 - Lokální optimalizace triangulace
- 8 Digitální model terénu
 - Lineární interpolace vrstevnic
 - Analýza sklonu
 - Analýza orientace

1. Tvorba digitálního modelu terénu



2. Formulace problému

Dáno: Množina bodů $P = \{p_1, p_2, \dots, p_n\}$ v \mathbb{R}^2 .

Hledáme: Triangulaci T nad množinou P .

Definice:

Triangulace \mathcal{T} nad množinou bodů P představuje takové planární rozdělení, které vytvoří soubor m trojúhelníků $t = \{t_1, t_2, \dots, t_m\}$ a hran tak, aby platilo:

- Libovolné dva trojúhelníky $t_i, t_j \in \mathcal{T}$, ($i \neq j$), mají společnou nejvýše hranu.
- Sjednocení všech trojúhelníků $t \in \mathcal{T}$ tvoří $\mathcal{H}(P)$.
- Uvnitř žádného trojúhelníku neleží žádný další bod z P .

Vztah mezi počtem bodů n , hran n_h a trojúhelníků n_t v rovině pro \mathcal{T}

$$n_h = 3n - 3 - k,$$

$$n_t = 2n - 2 - k,$$

k počet bodů ležících na $\mathcal{H}(P)$:

Odhad pouze funkcí n

$$n_h \leq 3n - 6,$$

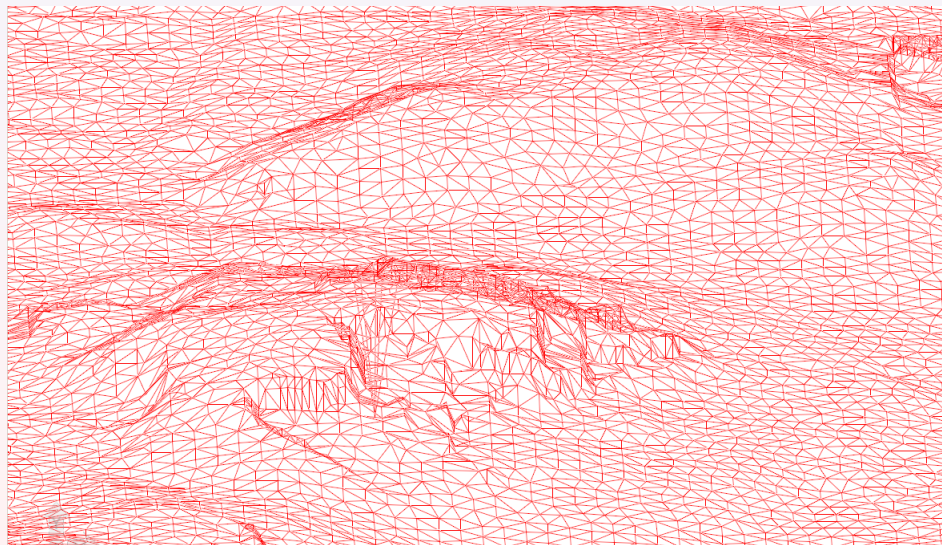
$$n_t \leq 2n - 5.$$

3. Použití triangulací

Aplikace triangulační algoritmů:

- Kartografie & GIS: tvorba digitálních modelů terénu.
- Zpracování obrazu: segmentace, rozpoznávání vzorů.
- Počítačová grafika: vizualizace prostorových dat ve scénách.
- FEM: analýza vlastností a struktury materiálů, simulace.
- Kartografická generalizace: liniové prvky, displacement.
- Plánování pohybu robotů: motion planning.
- Modelování přírodních jevů: eroze.
- Modelování pohybujících se objektů: kinematic triangulations.
- Biometrie (detekce otisků prstů, oční sítnice).
- Konstrukce dalších geometrických struktur (α -shapes).
- Teorie grafů: tvorba iniciálního grafu (TSP).

4. Rekonstrukce terénu z dat leteckého laserového skenování



5. Aplikace triangulací v biometrii

Detekce otisků prstů (Bebis et al., 2000)



6. Požadavky na triangulaci \mathcal{T}

Požadavky na triangulační algoritmus:

- Jednoduchost algoritmu, snadná implementace.
- Dostatečná rychlost pro velká P ($n > 1E6$) bodů, algoritmus $O(n \cdot \log(n))$.
- Malá citlivost na singulární případy (\mathcal{T} není jednoznačná).
- Převod do vyšších dimenzí: 3D.
- Schopnost paralelizace algoritmu.
- Optimální tvar trojúhelníkové sítě.

Některé body v kontrastu: jednoduchost implementace x rychlost.

Triangulační algoritmy patří mezi jedny z nejvíce teoreticky rozpracovaných.

V praxi používána téměř výhradně Delaunay triangulace.

7. Volba triangulace a jejich dělení

Při výběru \mathcal{T} nutno zohlednit další kritéria:

Tvar trojúhelníků:

Generování pravidelných trojúhelníků vhodných tvarů (blíží se rovnostranným).
Trojúhelníková síť se musí co nejvíce přimykát k terénu.
Kritérium je důležité při tvorbě DMT.

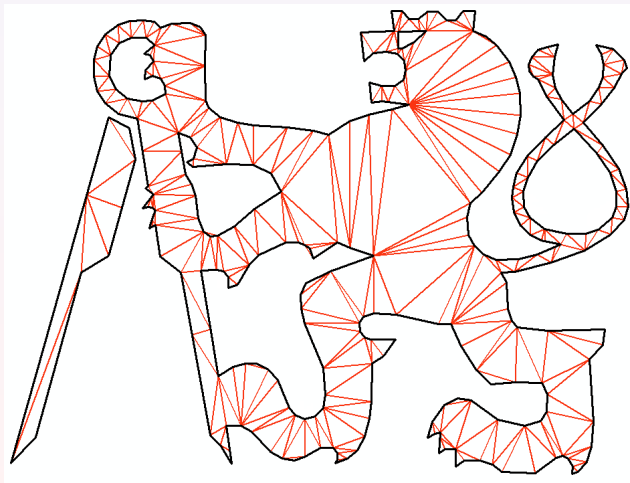
Povinné hrany:

Schopnost vkládat povinné hrany a modifikovat tvar triangulace.
Ovlivnění tvaru terénu, vkládání kosterních čar, tj. hřbetnic, údolnic, spádnic.

Triangulace nekonvexní oblasti:

Schopnost triangulovat nekonvexní oblasti či oblasti obsahující díry.
V mapách vynechány některé oblasti, např. vodní plochy, budovy.

8. Ukázka triangulace nekonvexní oblasti obsahující díry



9. Dělení triangulací

Dělení triangulací dle geometrické konstrukce:

- Greedy triangulace.
- Delaunay triangulace.
- MWT (Minimum Weight Triangulation).
- Constrained triangulace (triangulace s povinnými hranami).
- Datově závislé triangulace.

Dělení triangulací dle použitých kritérií:

- *Lokálně optimální triangulace*
Konvexní čtyřúhelník tvořený 2 sousedními trojúhelníky triangularizován optimálně vzhledem k zadanému kritériu.
- *Globálně optimální triangulace*
Všechny trojúhelníky triangulace \mathcal{T} optimální vzhledem k zadanému kritériu.
Globálně optimální triangulace je současně lokálně optimální.
- *Multikriteriálně optimalizované triangulace*
Kombinace několika lokálních či globálních kritérií.
Dlouhé výpočetní časy, doposud nejsou známy efektivní algoritmy, použití genetických algoritmů.

V praxi nejčastější globálně optimální triangulace.

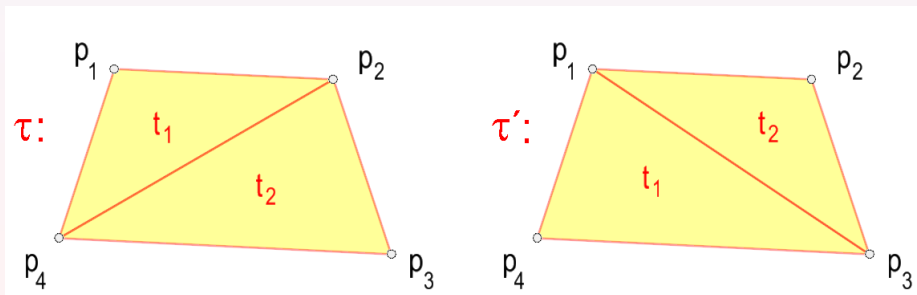
10. Hodnocení triangulace

Množina bodů $P = \{p_1, p_2, p_3, p_4\}$, $p_i \in \mathcal{H}$, konvexní čtyřúhelník.

Pak dle Eulerovy věty pro $n = 4$, $k = 4$ platí: $n_h = 5$, $n_t = 2$.

Důsledek: existují dvě různé triangulace $\mathcal{T}(P)$ a $\mathcal{T}'(P)$:

$$\mathcal{T}(P) = \{t_1(p_1, p_2, p_3), t_2(p_1, p_3, p_4)\}, \quad \mathcal{T}'(P) = \{t'_1(p_1, p_2, p_4), t'_2(p_2, p_3, p_4)\}.$$



Vhledem k posuzovanému kritériu je jedna z triangulací *optimální*, tj. minimalizuje ho.

12. Lokální/globální kritéria

Geometrický podtext, snaha o generování trojúhelníků “rozumných” tvarů.

Přehled nejčastěji používaných lokálních kritérií:

- Min/max úhel v trojúhelníku α : Delaunay triangulace maximalizuje minimální úhel.
- Min/max výška v trojúhelníku v .
- Min/max poloměr vepsané resp. opsané kružnice r resp. R .
- Min/max plocha trojúhelníku S .
- Úhel mezi normálami sousedních trojúhelníků: používáno u 2.5D triangulací

Přehled nejčastěji používaných globálních kritérií:

- *Suma délek stran*:
Minimalizace celkové délky hran h_i triangulace $\mathcal{T}(P)$

$$\sum_{i=1}^{n_h} h_i = \min.$$

MWT (Minimum Weight Triangulation), NP problém.

Přibližné řešení: genetické algoritmy, Greedy triangulace.

- *Povinné hrany*:
Předem definované hrany uvnitř triangulace, tzv. *Constrained Triangulation*.
 $\mathcal{T}(P)$ není lokálně optimální.
Zadání charakteristických hran terénních tvarů, lepší aproximace terénu.

13. MIN/MAX strategie, úhel v trojúhelníku

Vrcholy konvexního 4-úhelníku $P = \{p_i, p_j, p_k, p_l\}$

$$\mathcal{T}(P) = \{t_1(p_i, p_l, p_j), t_1(p_i, p_j, p_k)\}, \quad \mathcal{T}'(P) = \{t'_1(p_i, p_l, p_k), t'_2(p_l, p_j, p_k)\}.$$

Odstranění trojúhelníků s příliš ostrými/tupými úhly \Rightarrow tvarově nevhodné.

Min-max kritérium:

Eliminace trojúhelníků s příliš tupými úhly

$$\bar{\alpha} = \max_{\mathcal{T}}(\alpha), \quad \bar{\alpha}' = \max_{\mathcal{T}'}(\alpha').$$

Minimalizace maximálního úhlu v triangulaci

$$\mathcal{T} = \begin{cases} \mathcal{T}(P), & \bar{\alpha} \leq \bar{\alpha}', \\ \mathcal{T}'(P), & \bar{\alpha} > \bar{\alpha}'. \end{cases}$$

Max-min kritérium:

Eliminace trojúhelníků s příliš ostrými úhly

$$\underline{\alpha} = \min_{\mathcal{T}}(\alpha), \quad \underline{\alpha}' = \min_{\mathcal{T}'}(\alpha').$$

Minimalizace maximálního úhlu v triangulaci

$$\mathcal{T} = \begin{cases} \mathcal{T}(P), & \underline{\alpha} \geq \underline{\alpha}', \\ \mathcal{T}'(P), & \underline{\alpha} < \underline{\alpha}'. \end{cases}$$

14. Greedy triangulace

Patří do skupiny hladových algoritmů (Greedy Algorithms)

$$W(\mathcal{T}(P)) = \min \sum_{i=1}^{n_h} \|h_i\|_2.$$

Inkrementální konstrukce, přidáváme k -tou nejmenší hranu

$$h_{[n]} = \arg \min_{\forall h_i \in h \setminus h_{[1]}, \dots, \setminus h_{[n-1]}} (\|h_i\|_2).$$

Vlastnosti triangulace:

Pokud se v P nevyskytují hrany se stejnou délkou, je triangulace jednoznačná.

Snaží se však vytvářet trojúhelníky s nejkratšími stranami.

Trojúhelníky nemusí splňovat žádnou speciální geometrickou podmínku.

Jednoduchá implementace.

Složitost je $O(n^3)$, lze optimalizovat na $O(n^2 \cdot \log(n))$.

Důsledek:

Síť trojúhelníků není z tvarového hlediska optimalizována,

Do triangulace tak mohou být přidány tvarově nevhodné trojúhelníky.

V kartografii není příliš často používána.

Výsledná triangulace se blíží MWT.

15. Algoritmus Greedy triangulace

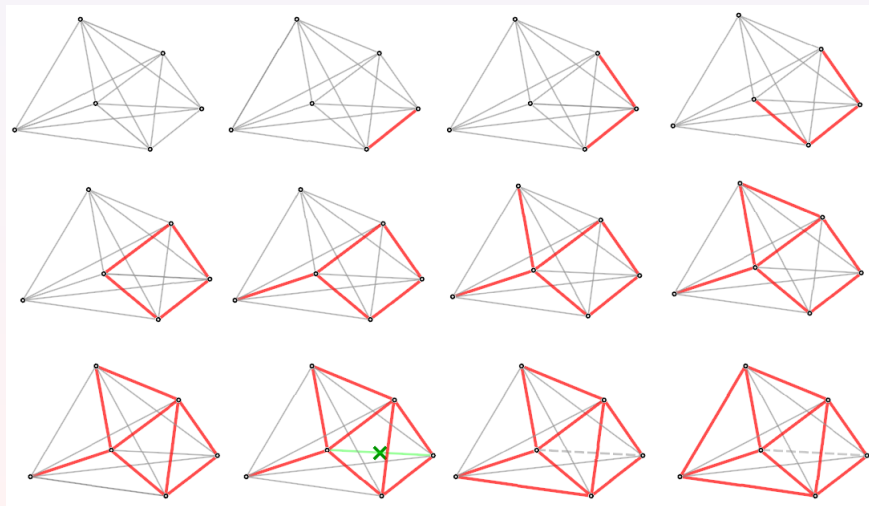
Využití prioritní fronty PQ .

Algoritmus 1: Greedy Triangulation (S, \mathcal{T})

```

1:  $PQ = \emptyset$ 
2: for  $\forall p_i, i \in \langle 1, n \rangle$ :
3:     for  $j \in \langle i + 1, n \rangle$ :
4:         Vytvoř hranu  $h = (p_i, p_j)$ .
5:          $PQ \leftarrow (\|h\|, h)$ .
6:  $\mathcal{T} \leftarrow Q.pop()$ .
7: while  $PQ$  not empty:
8:      $h = PQ.pop()$ 
9:      $intersect = false$ 
10:    for  $\forall h_i \in \mathcal{T}$ :
11:        if  $(h \cap h_i \neq \emptyset)$ 
12:             $intersect = true$ 
13:            break:
14:    if  $(!intersect)$   $\mathcal{T} \leftarrow h$ .
```


16. Grafické znázornění Greedy triangulace



17. Delaunay triangulace \mathcal{DT} a její vlastnosti

Nejčastěji používaná triangulace, v oblasti GIS de-facto standard.
Existuje v \mathbb{R}^2 i v \mathbb{R}^3 .

V1:

Uvnitř kružnice k opsané libovolnému trojúhelníku $t_j \in \mathcal{DT}$ neleží žádný jiný bod množiny P .

V2:

\mathcal{DT} maximalizuje minimální úhel v $\forall t$, avšak \mathcal{DT} neminimalizuje maximální úhel v t .

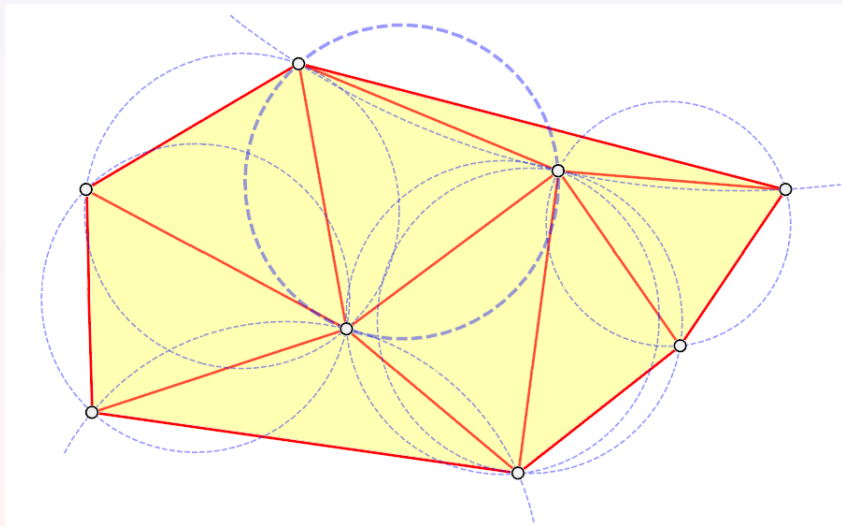
V3:

\mathcal{DT} je lokálně optimální i globálně optimální vůči kritériu minimálního úhlu.

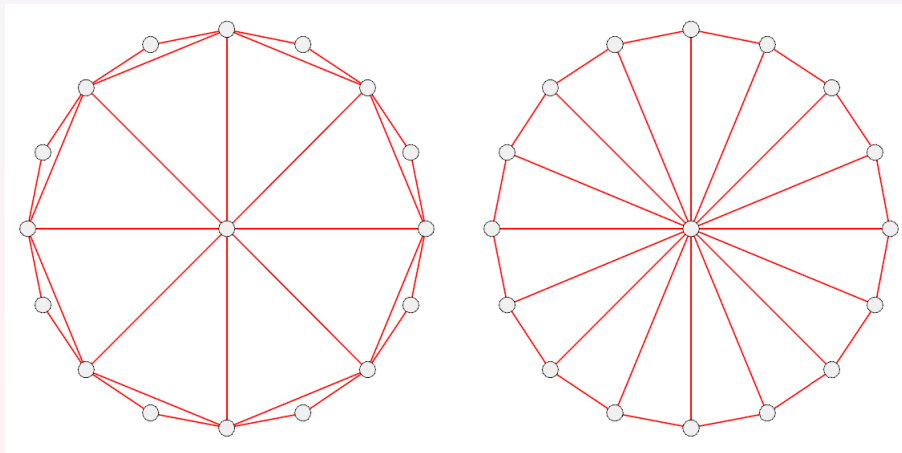
V4:

\mathcal{DT} je jednoznačná, pokud žádné čtyři body neleží na kružnici.

Výsledné trojúhelníky se při porovnání ze všemi známými triangulacemi nejvíce blíží rovnostranným trojúhelníkům.

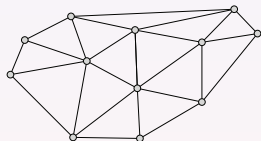
18. Ukázka DT 

19. Srovnání GT a DT

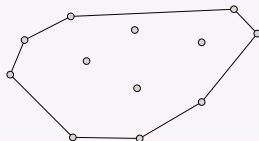


20. *DT* a podgrafy

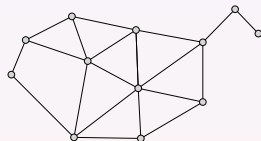
Důležité podgrafy Delaunay triangulace.



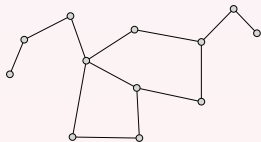
Delaunay Triangulation



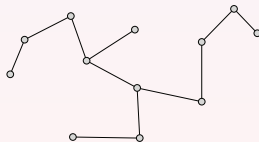
Convex Hull



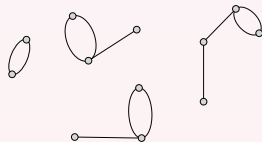
Gabriel Graph



Relative Neighbourhood Graph



Minimum Spanning Tree



Nearest Neighbourhood Graph

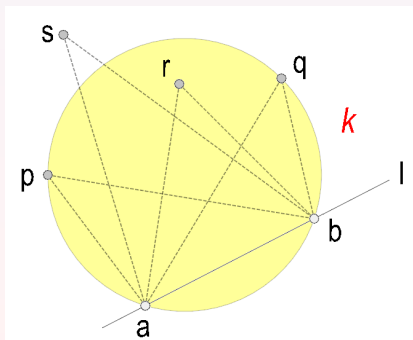
20. Geometrické vlastnosti DT

Nechť k je kružnice l přímka protínající k v bodech a, b .

Body p, q, r, s leží v polorovině σ_L .

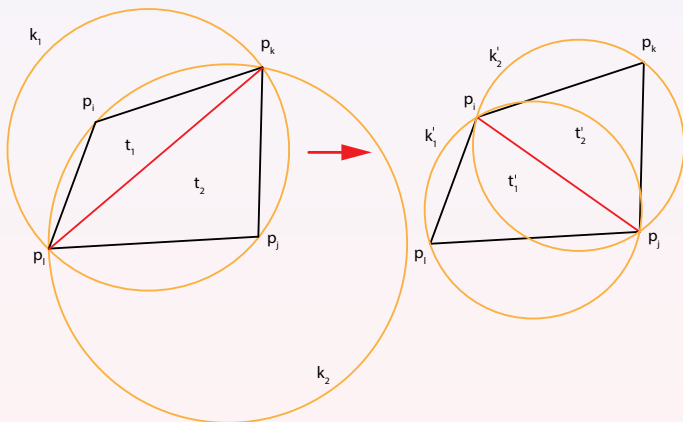
Důsledek Tháletovy věty

$$\angle arb > \angle apb = \angle aqb > asb.$$



21. Edge Flip, legalizace

Přechod $\mathcal{T}(P) \Rightarrow DT(P)$, prohození diagonály $(p_k, p_l) \Rightarrow (p_i, p_j)$.
 Triangulace legální, tj. lokálně optimální vzhledem k max-min kritériu.



Poloměry opsaných kružnic $k_1(S_1, r_1)$, $k_2(S_2, r_2)$ a $k'_1(S'_1, r'_1)$, $k'_2(S'_2, r'_2)$

$$r'_1 < r_1, \quad r'_2 < r_2.$$

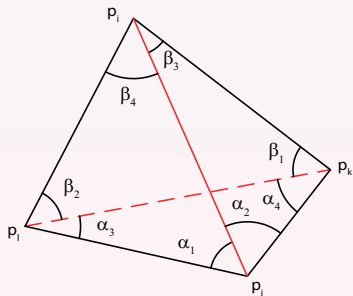
22. Vztah mezi úhly v $\mathcal{DT}(P)$ a $\mathcal{DT}'(P)$

Vnitřní úhly v trojúhelnících, $\mathcal{T}(P)$:

$$\alpha_1 + \alpha_2, \quad \alpha_3, \quad \alpha_4, \quad \beta_1, \quad \beta_2, \quad \beta_3 + \beta_4.$$

Vnitřní úhly v trojúhelnících, $\mathcal{DT}(P)$ (maximalizuje minimální úhel):

$$\alpha_1, \quad \alpha_2, \quad \beta_3, \quad \beta_4, \quad \beta_1 + \alpha_4, \quad \beta_2 + \alpha_3.$$



Pro každý úhel po swapu existuje nejméně jeden menší před swapem

$$\alpha_1 > \beta_1, \quad \alpha_2 > \beta_2, \quad \beta_3 > \alpha_3,$$

$$\beta_4 > \alpha_4, \quad \beta_1 + \alpha_4 > \alpha_4, \quad \beta_2 + \alpha_3 > \alpha_3.$$

Jinak řečeno, každý úhel se zvětší.

23. Nejednoznačnost DT

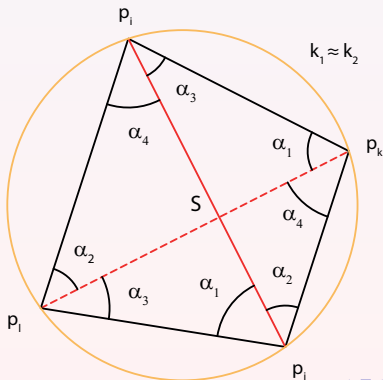
Pokud body $\{p_i, p_j, p_k, p_l\}$ na kružnici $k(S, r)$, pak $DT(P) \equiv DT'(P)$.

Vnitřní úhly v trojúhelnících, $DT(P)$:

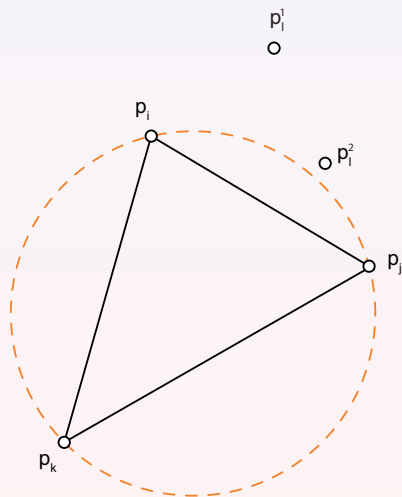
$$\alpha_1 + \alpha_2, \quad \alpha_3, \quad \alpha_4, \quad \alpha_1, \quad \alpha_2, \quad \alpha_3 + \alpha_4.$$

Vnitřní úhly v trojúhelnících, $DT'(P)$:

$$\alpha_1, \quad \alpha_2, \quad \alpha_3, \quad \alpha_4, \quad \alpha_1 + \alpha_4, \quad \alpha_2 + \alpha_3.$$

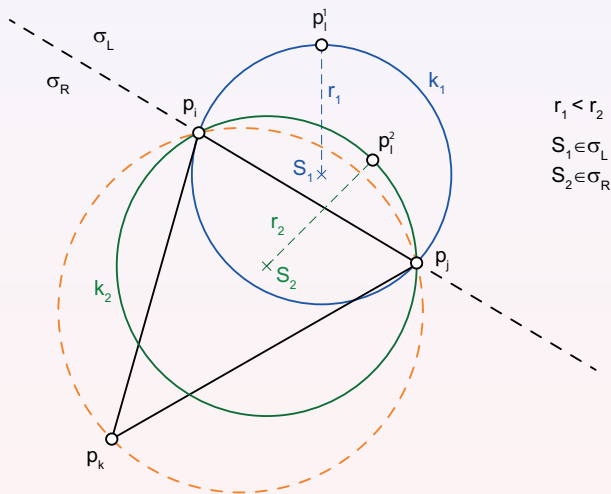


24. Který z bodů splňuje podmínku DT (1/2)?



Delaunay bod p negeneruje kružnici s minimálním poloměrem.

25. Který z bodů splňuje podmínku DT (2/2)?



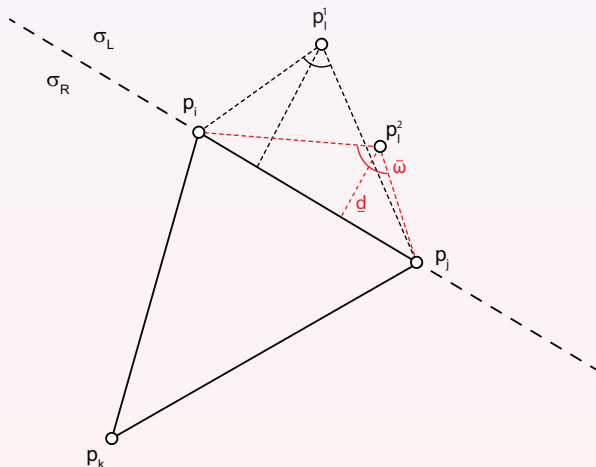
Ověření s využitím kritérií:

- Přímá (testy legality).
- Nepřímá (geometrická kritéria).

26. Geometrická kritéria

Geometrická kritéria:

- bod s minimální vzdáleností \underline{d} ke straně p_i, p_j ,
- bod s maximální úhlem $\bar{\omega}$ na stranu p_i, p_j .



27. Test legality I, opsaná kružnice

Vrcholy Delaunay trojúhelníku t : $p_1 = [x_1, y_1]$, $p_2 = [x_2, y_2]$, $p_3 = [x_3, y_3]$

Kružnice $k(p_1, p_2, p_3)$ opsaná t .

Analyzovaný bod $p = [x, y]$.

$p \in k$?

Předpoklad:

Vrcholy p_1, p_2, p_3 mají CW orientaci.

Testovací kritérium t :

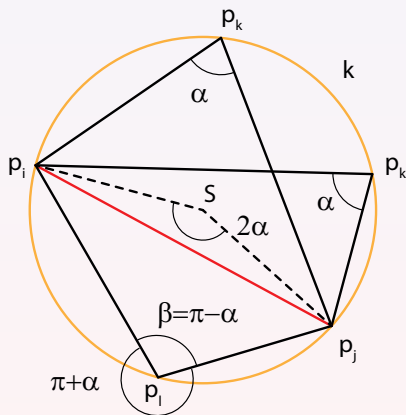
Determinant

$$t = \begin{vmatrix} x & y & x^2 + y^2 \\ x_1 & y_1 & x_1^2 + y_1^2 \\ x_2 & y_2 & x_2^2 + y_2^2 \\ x_3 & y_3 & x_3^2 + y_3^2 \end{vmatrix} \Rightarrow \begin{cases} t > 0, & p \notin k, \\ t = 0, & p \in \partial k, \\ t < 0, & p \in k. \end{cases}$$

Numericky stabilní.

Průmyslový test, pouze +, -, *.

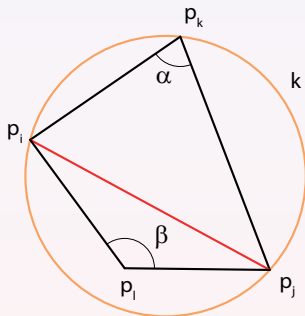
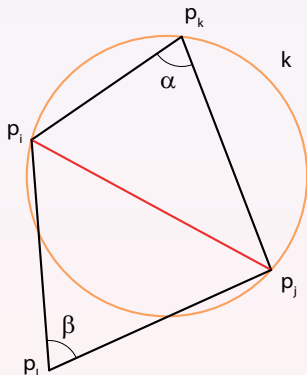
28. Úhly ve čtyřúhelníku (1/2)



$$\alpha + \beta = \alpha + \pi - \alpha = \pi.$$

29. Úhly ve čtyřúhelníku (2/2)

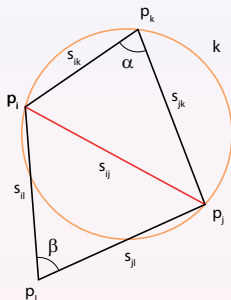
Legální vs. nelegální triangulace:



$$\alpha + \beta \begin{cases} < \pi, & DT(P) \text{ je legální,} \\ = \pi, & \text{Body } p_i, p_j, p_k, p_l \text{ leží na kružnici,} \\ > \pi, & DT(P) \text{ je nelegální.} \end{cases}$$

30. Test legality II., úhly

Numericky robustní test ověřující legalitu dvojice trojúhelníků $t_1(p_i, p_j, p_k)$ a $t_2(p_i p_j, p_l)$.



Nelegální swap

$$\sin(\alpha + \beta) = \cos \alpha \sin \beta + \cos \beta \sin \alpha < \sin(\pi).$$

Swapujeme, pokud

$$(x_{ik}x_{jk} + y_{ik}y_{jk})(x_{jl}y_{il} - x_{il}y_{jl}) < (x_{jk}y_{ik} - x_{ik}y_{jk})(x_{jl}x_{il} + y_{jl}y_{il}),$$

kde

$$x_{ik} = x_i - x_k, \quad x_{jk} = x_j - x_k, \quad x_{jl} = x_j - x_l, \quad x_{il} = x_i - x_l,$$

$$y_{ik} = y_i - y_k, \quad y_{jk} = y_j - y_k, \quad y_{jl} = y_j - y_l, \quad y_{il} = y_i - y_l.$$

Test numericky stabilní, pouze +, -, *.

31. Metody konstrukce \mathcal{DT}

Metody přímé konstrukce \mathcal{DT} :

- Lokální prohazování: $O(n)$ + triang. algoritmus.
- Inkrementální konstrukce: $O(n^3)$.
- Inkrementální vkládání: $O(n \log n) - O(n^2)$.
- Divide and conquer: $O(n \log n)$, např. DeWall.
- Sweep Line: $O(n^2)$.

Nepřímá konstrukce:

Přes Voronoi diagram, v praxi není používána.

Velká data (point clouds):

Inkrementální konstrukce nebo D&C (DeWall).

Ostatní metody příliš pomalé.

Metoda lokálního prohazování:

Metoda je použitelná pouze ve 2D, obtížně lze převést do vyšší dimenze.

Převod libovolné triangulace \mathcal{T} na \mathcal{DT} .

Prohazování nelegálních hran v dvojicích trojúhelníků tvořících konvexní čtyřúhelník.

Složitost algoritmu je $O(n)$, nutno připočítat složitost na triangulačního algoritmu.

Lze použít vzhledem k libovolnému kritériu.

32. Algoritmus lokálního prohozování

Algoritmus 2: Delaunay Triangulation Local(P)

```

1: Vytvoř pomocnou triangulaci  $T(P)$ .
2: legal=false
3: while  $T(P)$  !legal
4:     legal=true;
5:     Opakuj pro  $\forall e_i \in T(P)$ 
6:         Vezmi hranu  $e_i \in T(P)$ 
7:         Nalezni trojúhelníky  $t_1, t_2$  incidující s  $e_i$ .
8:         if  $(t_1 \cup t_2)$  konvexní a nelegální
9:             Legalize  $(t_1, t_2)$ .
10:            legal=false;

```

33. Metoda inkrementální konstrukce

Algoritmus lze použít ve 2D (prázdná kružnice) i 3D (prázdná sféra).
Založena na postupném přidávání bodů do již vytvořené \mathcal{DT} .

Aktuální Delaunayovská hrana $e = (p_1, p_2)$.

V trojúhelníku orientovány CCW.

Hledán Delaunay point \bar{p} , maximalizující $\angle(p_1, p, p_2)$

$$\bar{p} = \arg \max_{\forall p_i \in \sigma_L(e)} \angle(p_1, p_i, p_2), \quad p_i \in \sigma_1(e).$$

Do \mathcal{DT} následně přidány hrany trojúhelníku $\triangle(p_1, p_2, \underline{p})$:

$$e_1 = (p_2, \underline{p}), e_2 = (\underline{p}, p_1),$$

pokud hrany $e'_1 = (p, p_2)$, $e'_2 = (p_1, p)$ s protilehlou orientací nejsou v AEL.

Při konstrukci používána datová struktura **Active Edge List** (AEL):

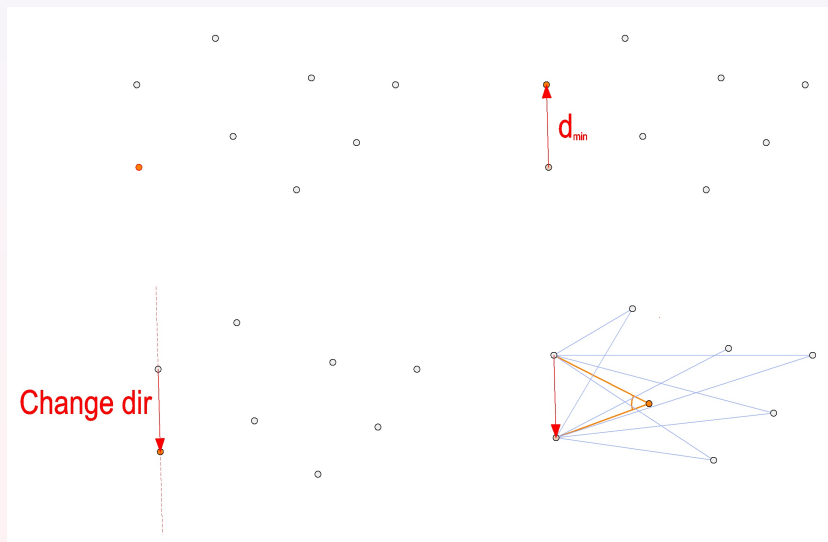
Obsahuje hrany e , ke kterým hledáme body \bar{p} .

Neukládá se topologický model.

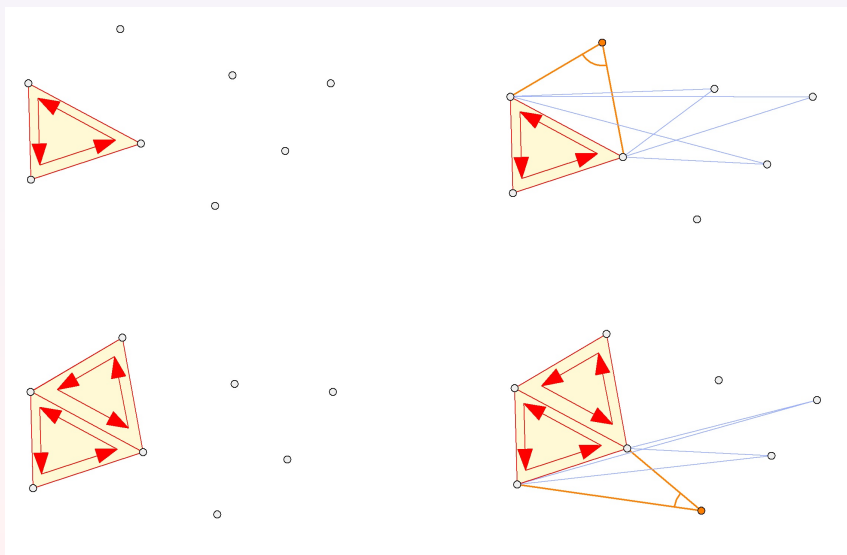
Složitost je $O(n^3)$, lze vylepšit.

Algoritmus nestabilní, numerické problémy (použití adaptivní matematiky).

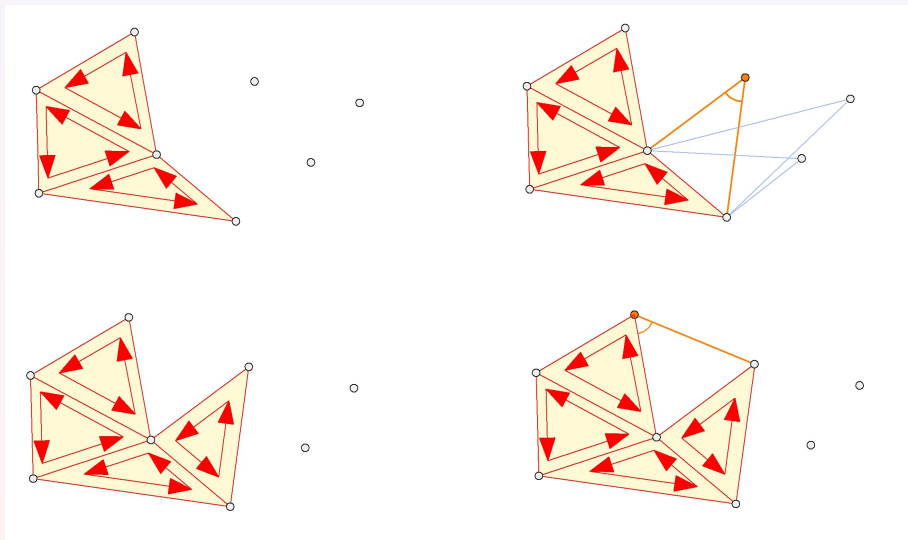
34. Ilustrace inkrementální konstrukce (1/3)



35. Ilustrace inkrementální konstrukce (2/3)



36. Ilustrace inkrementální konstrukce (3/3)



37. Algoritmus inkrementální konstrukce DT (1/2)

Algoritmus 2: Delaunay Triangulation Incremental (P, AEL, DT)

- 1: $AEL = \{\}, DT = \{\}$
 - 2: $p_1 = rand(P)$ //Náhodný bod
 - 3: $p_2 = \arg \min_{p_i \in P} \|p_1 - p_i\|_2$ //Nejbližší bod
 - 4: Vytvoř hrany $e = (p_1, p_2), e' = (p_2, p_1)$
 - 5: $AEL \leftarrow e, AEL \leftarrow e'$ //Pridani 2 hran do AEL
 - 6: while AEL not empty:
 - 7: $e_1 = AEL.pop(), e_1 = (p_1, p_2)$ //Vezmi první hranu z AEL
 - 8: $e'_1 = (p_2, p_1)$ //Prohod její orientaci
 - 9: $\bar{p} = \arg \max_{p_i \in \sigma_L(e'_1)} \angle(p_1, p_i, p_2)$ //Najdi Delaunayovský bod
 - 10: if $\exists \bar{p}$: //Takový bod existuje
 - 11: $e_2 = (p_2, \bar{p}), e_3 = (\bar{p}, p_1)$ //Vytvor zbyvajících hrany trojúhelníku
 - 12: $DT \leftarrow e'_1, DT \leftarrow e_2, DT \leftarrow e_3$ //Pridej hrany do DT
 - 13: updateAEL(e_2, AEL), add(e_3, AEL) //Update AEL
-

38. Algoritmus inkrementální konstrukce DT (2/2)

Při přidání $e = (a, b)$ do AEL.

Kontrola, zda neobsahuje hranu s opačnou orientací $e' = (b, a)$.

Pokud ano, je e' odstraněna z AEL.

Pokud ne, je e přidána do AEL.

Triangulace ukládána po trojúhelnících.

Algoritmus 3: updateAEL ($e = (a, b)$, AEL)

1: Vytvoř hranu $e' = (b, a)$ //Zmen orientaci

2: if ($e' \in AEL$)

3: $AEL \rightarrow e'$ //Odstran z AEL

4: else:

5: $AEL \leftarrow e$ //Pridej do AEL

V praxi není používán z důvodu složitosti pro velké množiny (hledání, mazání).

Výhodou je však poměrně jednoduchá implementace.

Pro rychlé hledání použita množina.

39. Metoda inkrementálního vkládání

Často používaná metoda konstrukce \mathcal{DT} .

Lze použít v \mathbb{R}^2 i \mathbb{R}^3 .

Složitost je $O(n^2)$, po úpravách lze dosáhnout $O(n \cdot \log(n))$.

Klasický případ rekurzivní úlohy.

Vícenásobná legalizace.

Princip algoritmu:

V každém kroku do \mathcal{DT} přidán jeden bod p .

Následně provedena aktualizace \mathcal{DT}

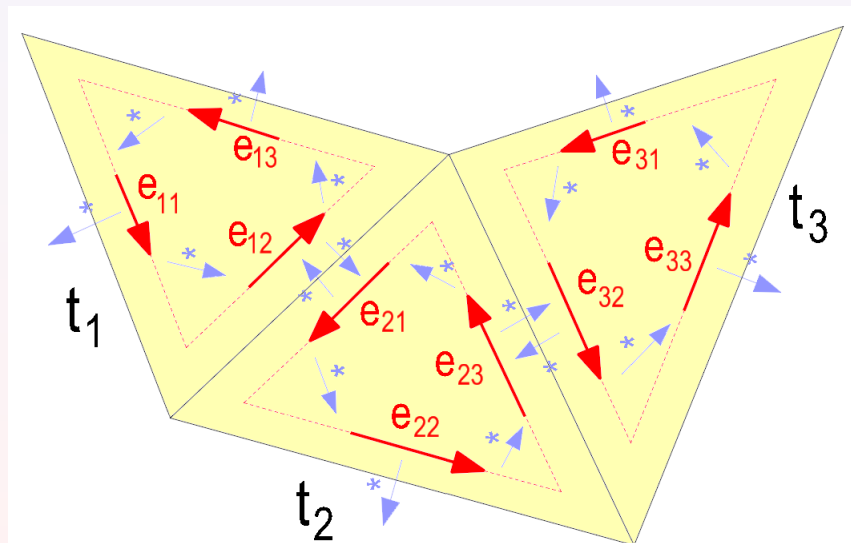
$$\mathcal{DT}(P_n) = \mathcal{DT}(P_{n-1}) + \Delta\mathcal{DT}(p).$$

Bod vybírán náhodně nebo inteligentně (co nejbliže od předchozího).

Algoritmus tvořen 4 fázemi:

- Konstrukce simplexu Ω oblasti P (obalový trojúhelník).
- Přidání p do \mathcal{DT}_m .
- Legalizace triangulace \mathcal{DT}_{m+1} .
- Odstranění simplexových hran.

40. Ukázka datového modelu



41. Fáze1: Konstrukce simplexu Ω

Žádný z bodů P neleží vně simplexu Ω s vrcholy p_{-1}, p_{-2}, p_{-3} .
 \mathcal{DT} bude probíhat nad sjednocením obou množin, tj. $\mathcal{DT}(P \cup \Omega)$.

Zaručíme tak, že přidání každého bodu p_i do \mathcal{DT} proběhne v souladu s souladu s níže uvedenými pravidly.

Vrcholy simplexu p_{-1}, p_{-2}, p_{-3} dostatečně daleko od P .
Co nejméně ovlivňují trojúhelníky nad body množiny P .

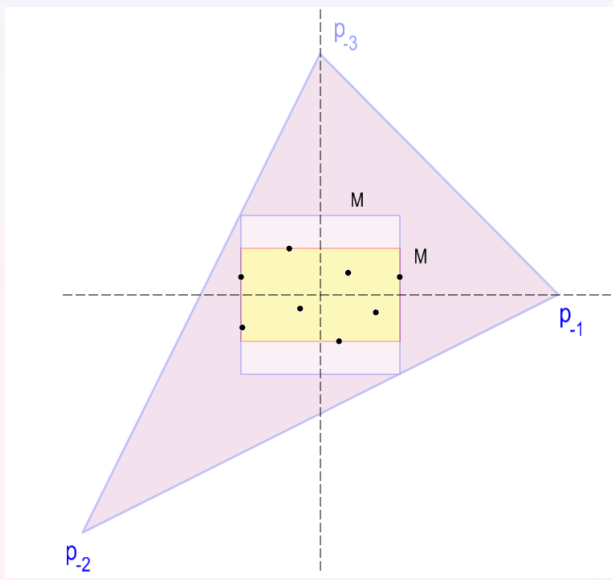
Souřadnice vrcholů simplexu Ω vcházejí z MBR nad P

$$M = \max(x_{max} - x_{min}, y_{max} - y_{min}).$$

Pak $p_{-1} = [-3M, 0]$, $p_{-2} = [0, 3M]$, $p_{-3} = [-3M, -3M]$.

Při legalizaci nutno upravit pravidla pro *simplexové trojúhelníky*.
Alespoň jeden jejich vrchol $p_i \in \Omega$.

42. Ilustrace simplexu Ω



43. Fáze 2: Přidání p do \mathcal{DT}

Nalezení trojúhelníku/trojúhelníků t_i , se kterými p inciduje.

Kritická pasáž algoritmu, výpočetně nejnáročnější krok.

Nelze prohledávat všechny trojúhelníky.

Množství procházených trojúhelníků nutno minimalizovat.

Vyhledání incidujícího trojúhelníku:

Dvě strategie:

- Metoda procházky (heuristika, $O(\sqrt{n})$), Point Location Problem.
- DAG Tree (konstrukce ternárního stromu, efektivnější, $O(\log_2 n)$).

Metody procházek:

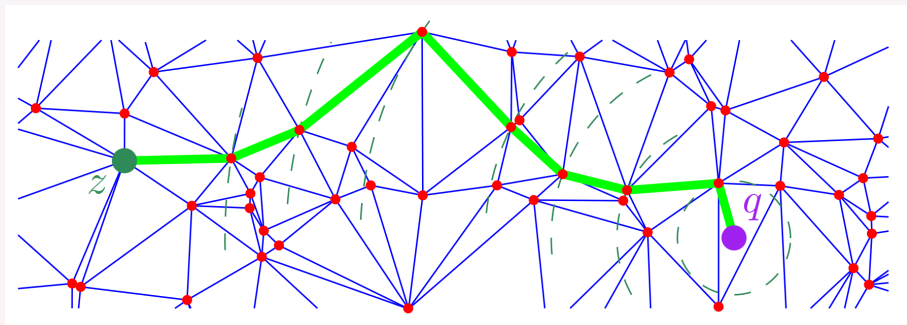
Startovní bod z , analyzovaný bod q .

Hledáme $t_i, q \in t_i$.

- Greedy Walk: přes nejbližší vrcholy.
- Straight Walk: přes t protnuté (q, z) .
- Visibility Walk: díváme se vlevo/vpravo.
- Orthogonal Walk: 2 ortogonální směry.

44. Greedy Walk

Nejjednodušší procházkový algoritmus.
Procházka přes sousedící trojúhelníky.
V každém okamžiku zvolen bod nejbližší od q .
Robustnost vůči numerickým chybám.



(Devillers, 2016).

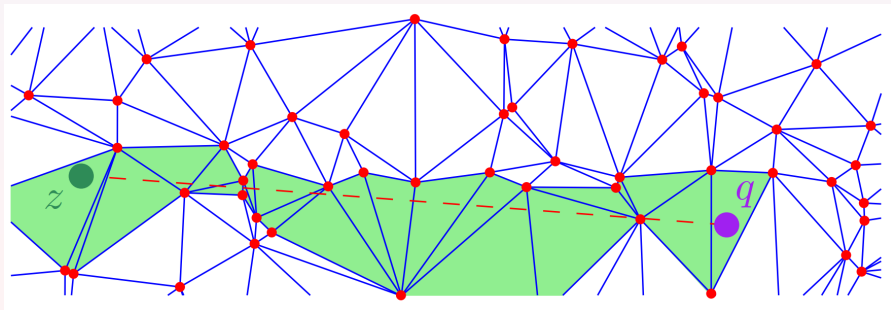
45. Straight Walk

Postupná cesta přes sousedící trojúhelníky protnuté (q, z) .

V každém trojúhelníku hledána protnutá strana.

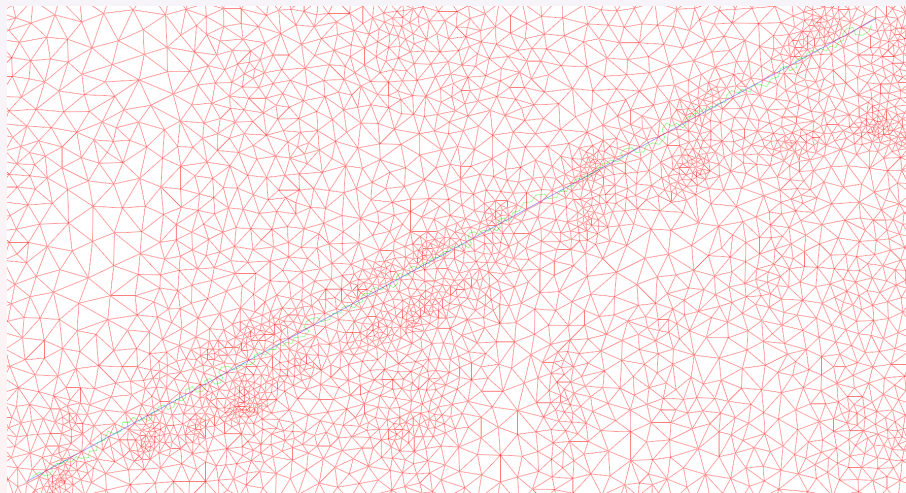
Alternativně orientační test.

Pozor na singularity: $p_i \in (q, z)$.



(Devillers, 2016).

46. Straight Walk



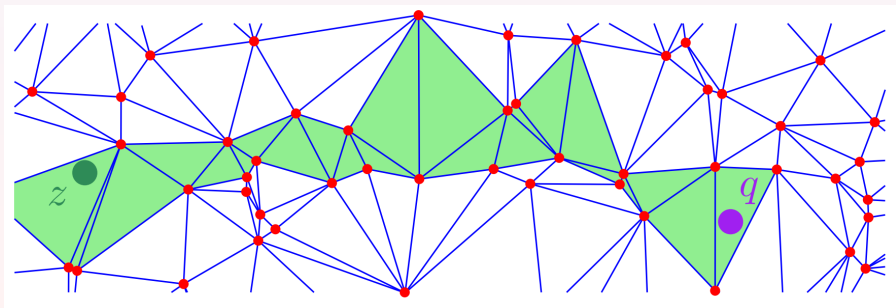
47. Visibility Walk

Postupná cesta přes sousedící trojúhelníky.

Nejefektivnější procházka, Lawson (1977).

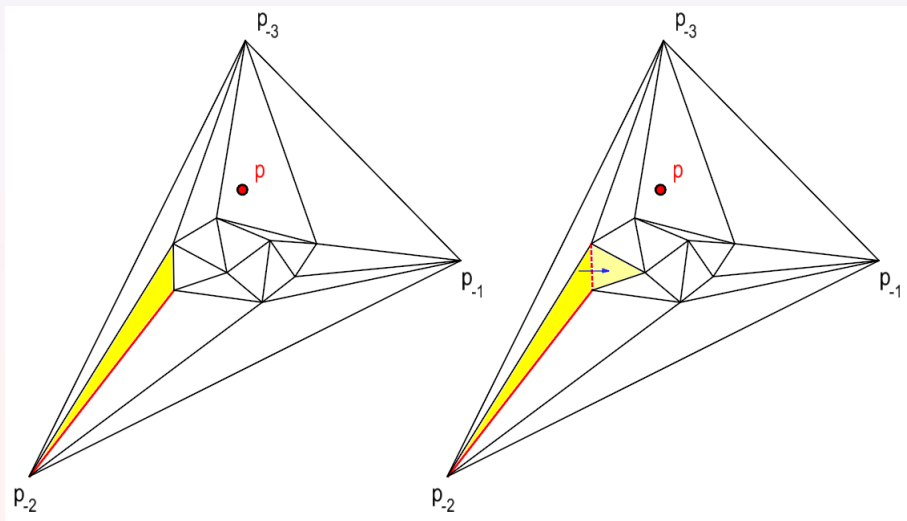
Z t_i přecházíme na t_j , pokud t_i a q v opačné polorovině vzhledem k hraně e_{ij}

$$e_{i,j} = \begin{cases} e_{i+1,j}, & q \in \sigma_l(e_{ij}), \\ e_{j,l}, & q \in \sigma_r(e_{ij}). \end{cases}$$

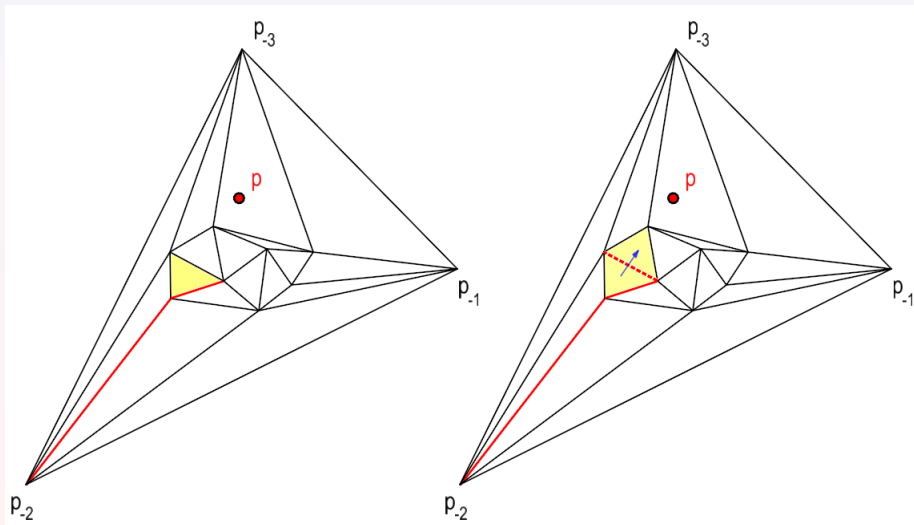


(Devillers, 2016).

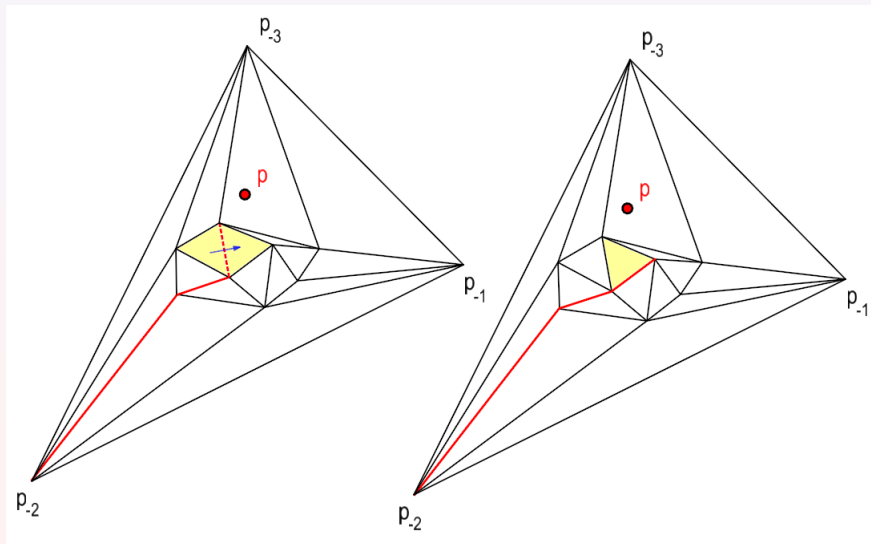
48. Ukázka Visibility Walk (1/6)



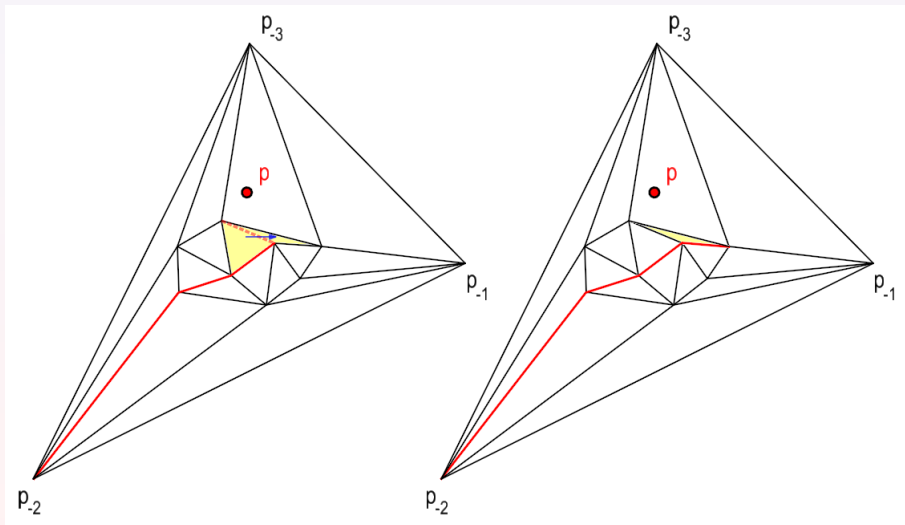
49. Ukázka Visibility Walk (2/6)



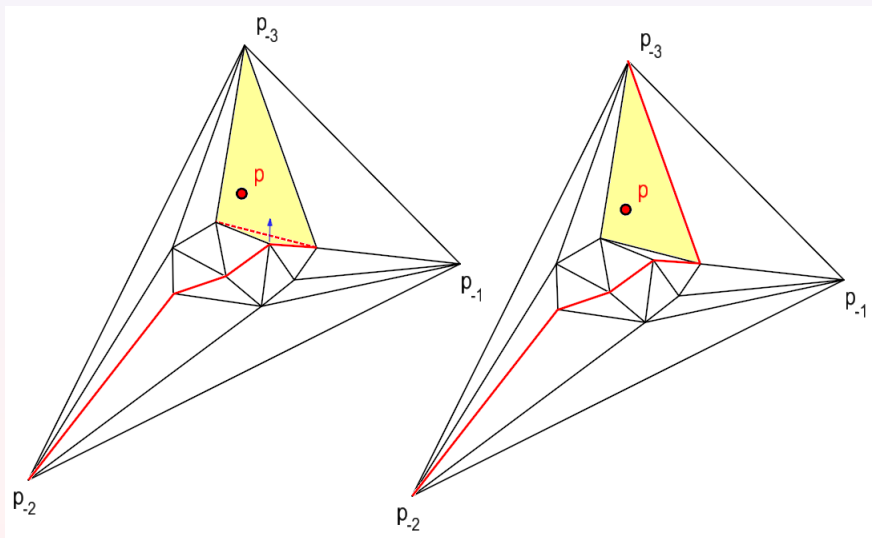
50. Ukázka Visibility Walk (3/6)



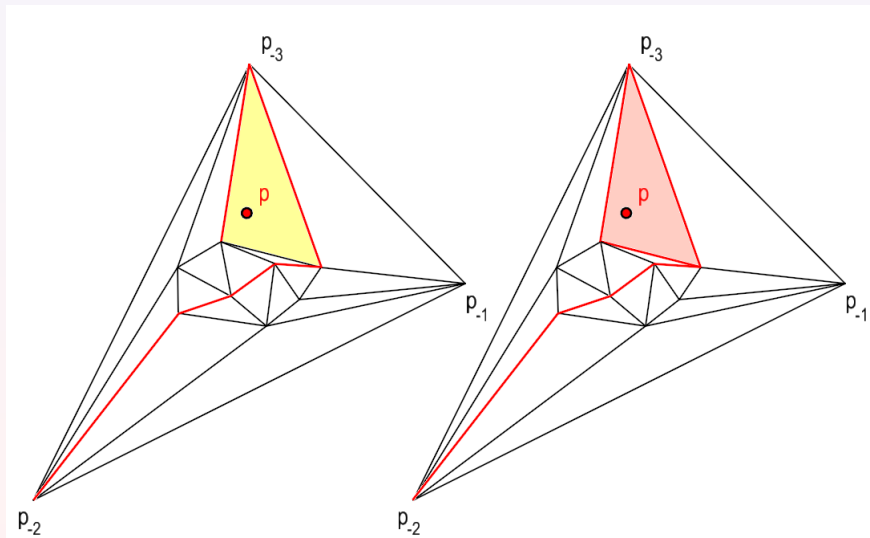
51. Ukázka Visibility Walk (4/6)



52. Ukázka Visibility Walk (5/6)



53. Ukázka Visibility Walk (6/6)



54. Visibility Walk, algoritmus

Složitost $O(\sqrt{n})$.

Point Location problem.

Pozor na float aritmetiku: pravidelné rastry.

Algoritmus 3: VW (p, τ, \cdot)

```

1:  $t = t_0$ , found = false
2: while (!found)
3:     found = true;
4:     for  $\forall e(a, b) \in \tau$  //Hrana  $e_{ij} = t \cap t_{inc}$ 
5:         if  $p \in \sigma_r(a, b)$ 
6:              $t \leftarrow t_{inc}$  //Incidující trojúhelník
7:             found = false
8:             break
9: return  $t$ 

```

55. Vztah bodu p a nalezeného t_i

Existují tři různé varianty vzájemné polohy přidávaného bodu p a nalezeného trojúhelníku t_i :

- *Bod p leží ve vrcholu t_i*

Bod neovlivní již vytvořenou triangulaci $\mathcal{DT}(P_{n-1})$, bude zanedbán.
Triangulace ponechána beze změny, $\mathcal{DT}(P_n) = \mathcal{DT}(P_{n-1})$.

- *Bod $p \in t_i$*

Zkonstruovány tři nové hrany spojující p s vrcholy t_i .

Trojúhelník t se rozpadne na tři nové trojúhelníky se společným vrcholem.

- *Bod p leží ve straně t_i, t_j*

Dva incidující trojúhelníky t_i, t_j , v jejichž společné hraně přidávaný bod leží.

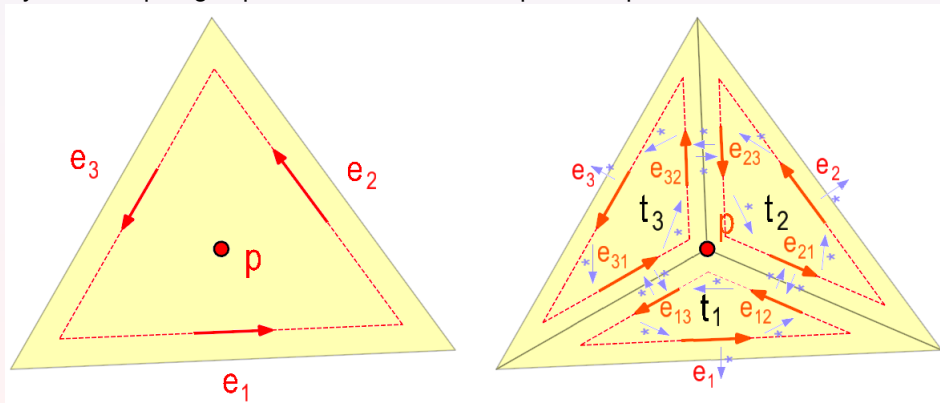
Rozděleny dvojicí úseček jdoucích z p do protilehlých vrcholů t_i, t_j .

Vzniknou čtyři nové trojúhelníky se společným vrcholem.

56. Bod $p \in t_i$

Vytvoření nových hran: $e_{12}, e_{13}, e_{21}, e_{23}, e_{31}, e_{32}$.

Vytvoření topologie: provázání všech hran s použitím pointerů.



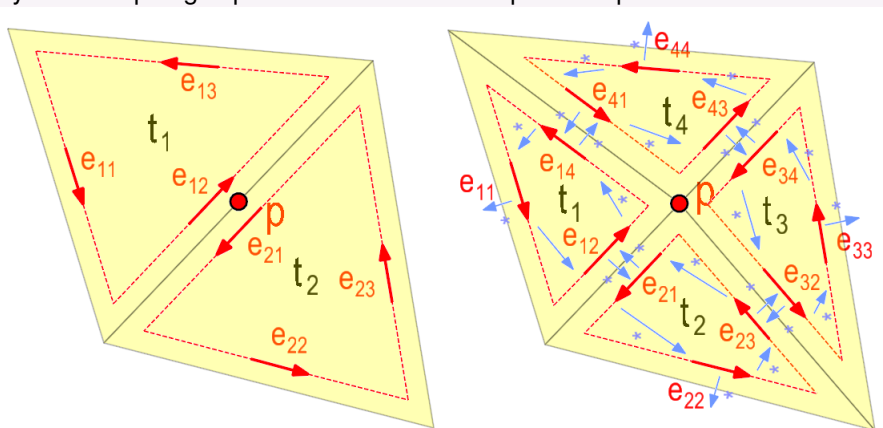
57. Bod p leží ve straně t_i, t_j

Změna koncových bodů hran: e_{12}, e_{21}, e_{23} .

Zrušení hrany e_{13} .

Vytvoření nových hran: $e_{14}, e_{32}, e_{33}, e_{34}, e_{41}, e_{43}, e_{44}$.

Vytvoření topologie: provázení všech hran s použitím pointerů.



58. Fáze 3: Legalizace nově vytvořené triangulace

Takto vzniklá triangulace nemusí být delaunayovská, triangulaci je proto nutno legalizovat.

- $p \in t_i$
Legalizujeme nově vzniklé trojúhelníky t_1, t_2, t_3 s incidujícími trojúhelníky \mathcal{DT}
Celkem 3 rekurzivní legalizace.
- *Bod p leží ve straně t_i, t_j*
Legalizujeme nově vzniklé trojúhelníky t_1, t_2, t_3, t_4 s incidujícími trojúhelníky \mathcal{DT} .
Celkem 4 rekurzivní legalizace.

Přehození úhlopříčky v některém z 4-úhelníku:

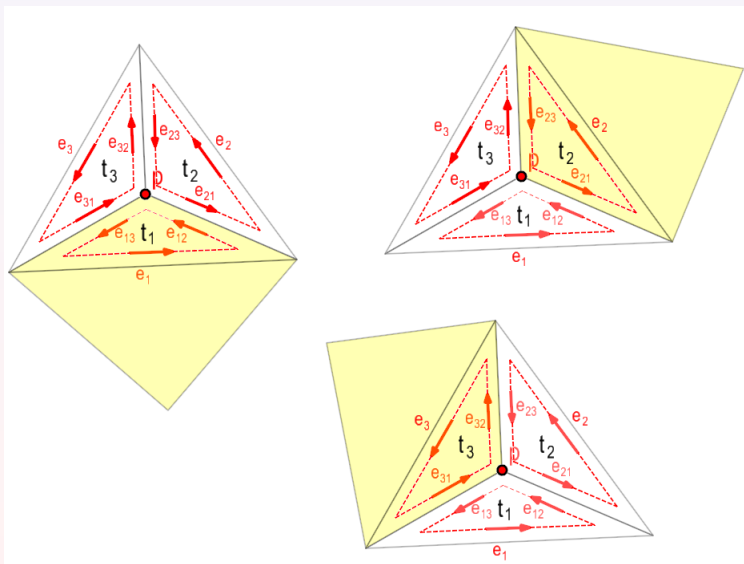
Vyvolá potřebu legalizace k jejich incidujícím trojúhelníkům.

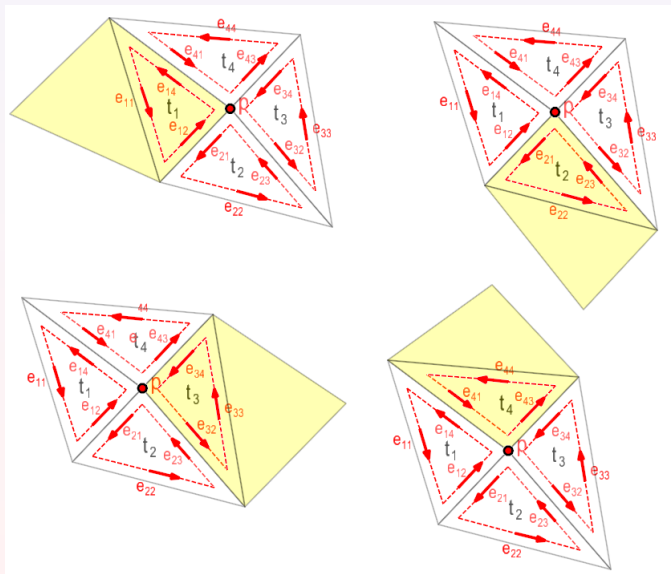
Pokud alespoň jeden z vrcholů trojúhelníka představuje bod Ω :

Nutno upravit legalizační pravidla, zde neswapujeme.

Tyto kroky vyvolají potřebu rekurzivního řešení problému.

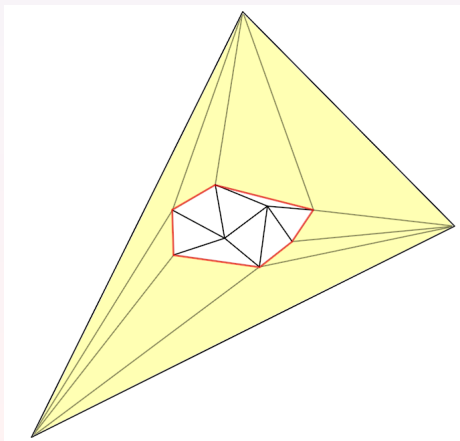
V nepříznivém případě může p způsobit přegenerování více t v \mathcal{DT} .

59. Ilustrace procesu legalizace, $p \in t_i$ 

60. Ilustrace procesu legalizace, p leží ve straně t_i, t_j 

61. Fáze 4: Odstranění simplexových hran

Odstranění všech stran $\mathcal{DT}(P \cup \Omega)$ které incidují z Ω , výsledkem $\mathcal{DT}(P)$.
Výsledkem oříznutí na konvexní obálku.



62. Triangulace se vstupní podmínkou

Označovány jako Constrained Triangulations (tj. triangulace s omezením).

Do triangulace zavedeny povinné hrany spojující definované body p .

Poloha povinných hran se při triangulaci již nesmí měnit.

Globální optimalita:

Povinné hrany při konstrukci triangulace kříží jiné “lepší” hrany.

Ty jsou vzhledem k nějakému kritériu lokálně optimální (a pro triangulaci vhodnější).

Avšak tyto hrany nejsou použity, jsou nahrazeny povinnými.

Triangulace se vstupní podmínkou proto nejsou lokálně ani globálně optimální.

Geometrický předpoklad:

Povinné hrany se nesmějí protínat ani být kolineární.

V takovém případě rozložení na více povinných hran.

Široké použití v kartografii tvorbě digitálních modelů terénu.

Povinné hrany umožňují lepší modelování morfologie terénu.

Zástupci:

- Greedy triangulace se vstupní podmínkou (Constrained Greedy Triangulation).
- Delaunay triangulace se vstupní podmínkou (Constrained Delaunay Triangulation).

V praxi používána pouze CDT.

63. Greedy triangulace se vstupní podmínkou

Greedy triangulace se vstupní podmínkou $\mathcal{CGT}(P)$ není příliš často používána.

Tvorba $\mathcal{CGT}(P)$ probíhá ve dvou krocích:

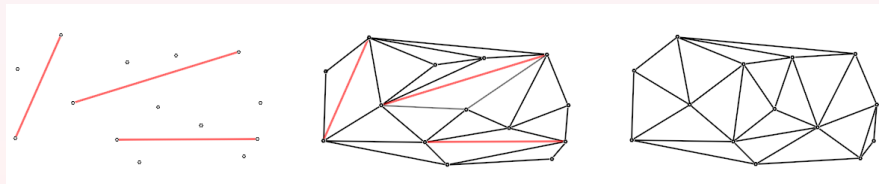
- *Přidání povinných hran*

Do prázdné triangulace přidány povinné hrany.

Žádná z povinných hran nemůže být při triangulaci nahrazena možnou kratší stranou.

- *Tvorba $\mathcal{GT}(P)$*

Konstrukce Greedy triangulace nad P .



64. Delaunay triangulace se vstupní podmínkou

Nejpoužívanější triangulace v geoinformatice.

Na rozdíl od $\mathcal{CGT}(P)$ nutná redefinice triangulace: přes povinné hrany neprobíhá swapování.

Triangulace jako celek nemaximalizuje minimální úhel v trojúhelnících.

Zavedení povinných hran sníží rychlost triangulačního algoritmu.

Možné geometrické problémy, povinná hrana:

- *Kolineární s nějakou hranou $\mathcal{DT}(P)$* :
Rozdělení povinné hrany na 3 části.
- *Povinná hrana protíná bod $\mathcal{DT}(P)$* :
Rozdělení povinné hrany na 2 části.

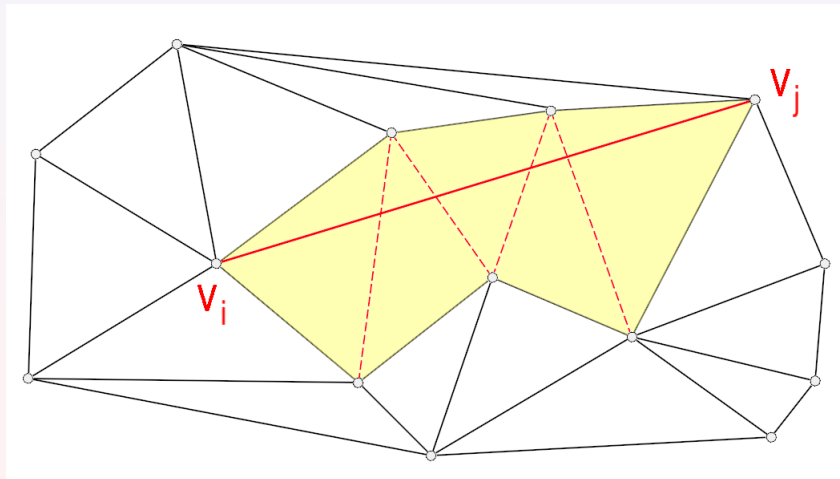
Konstrukce probíhá ve třech krocích:

- Vytvoření $\mathcal{DT}(P)$.
- Zadání povinných hran do $\mathcal{DT}(P)$.
- Převod $\mathcal{DT}(P)$ na $\mathcal{CDT}(P)$.

Pro bod 3 existuje řada algoritmů, např. Sloan (1992):

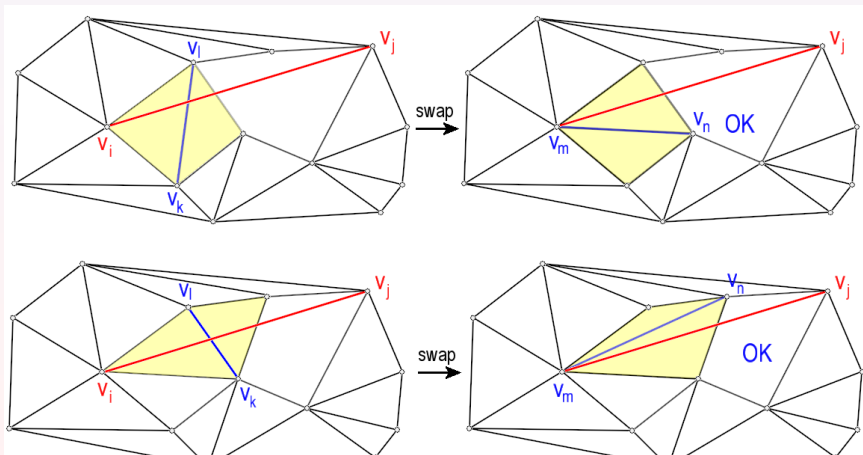
- 1 Nalezení stran $\mathcal{DT}(P)$ protínajících povinnou hranu (v_i, v_j) .
- 2 Zrušení všech stran v $\mathcal{DT}(P)$ protínajících povinnou hranu (v_i, v_j) .
- 3 Vytvoření pomocné triangulace.
- 4 Obnovení $\mathcal{DT}(P)$.
- 5 Odstranění nadbytečných trojúhelníků.

65. Nalezení stran $\mathcal{DT}(P)$ protínajících povinnou hranu (v_i, v_j)



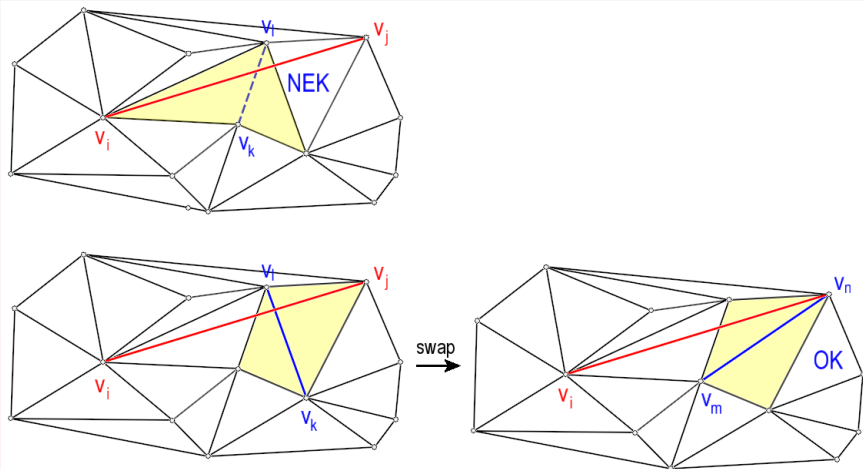
66. Ilustrace odstranění stran protínajících povinnou hranu (v_i, v_j) (1/3)

Tvoří incidující trojúhelníky protnuté hrany konvexní 4-úhelník?
Pokud ano, prohoď úhlopříčku, jinak pokračuj další protnutou hranou.

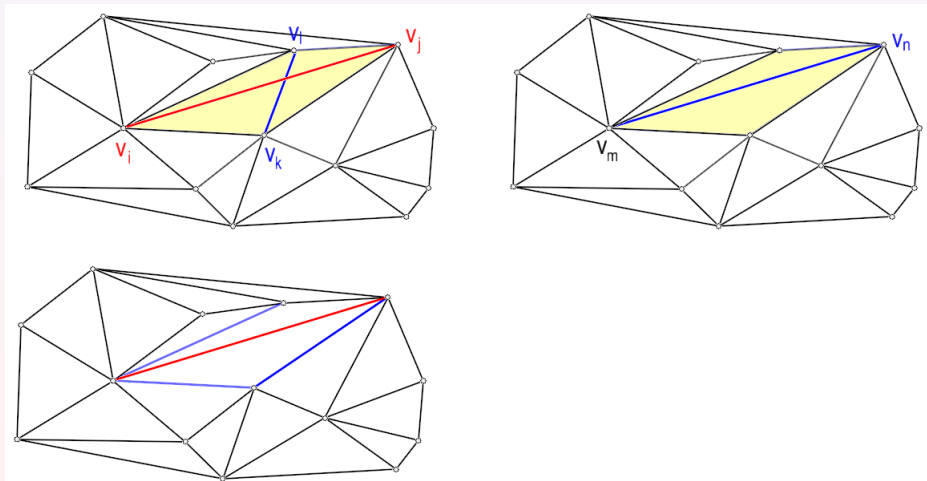


67. Ilustrace odstranění stran protínajících povinnou hranu (v_i, v_j) (2/3)

V konečném počtu kroků vytvořena prozatímní triangulace.



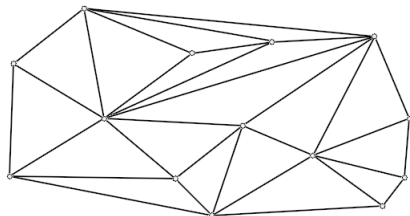
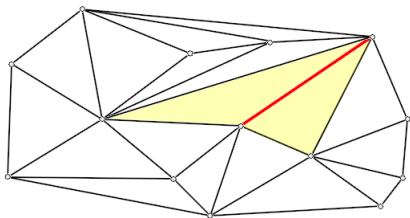
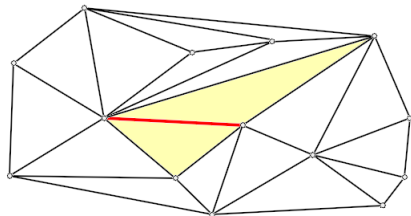
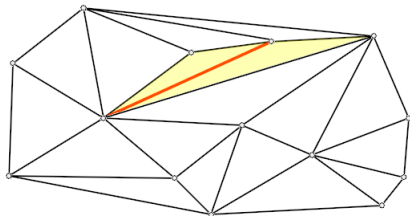
68. Ilustrace odstranění stran protínajících povinnou hranu (v_i, v_j) (3/3)



69. Ilustrace obnovení $DT(P)$

Priztatimní triangulace není Delaunayovská.

Provedena legalizace vzhledem k incidujícím trojúhelníkům: prohazování hran.



70. Triangulace nekonvexní oblasti a oblasti s otvory

Triangulace nekonvexní oblasti:

Triangulace množiny bodů ohraničené nekonvexním polygonem.

Zpravidla použita CDT, povinné hrany = polygon.

Z triangulace odstraněny všechny trojúhelníky vně polygonu.

Pro trojúhelník určeno těžiště.

Pokud vně polygonu, trojúhelník odstraněn (PLP).

Typické u DMT, triangulace nekonvexní oblasti se vstupními daty.

Vně oblasti by si algoritmus DMT “vymýšlel”.

Triangulace oblastí s otvory:

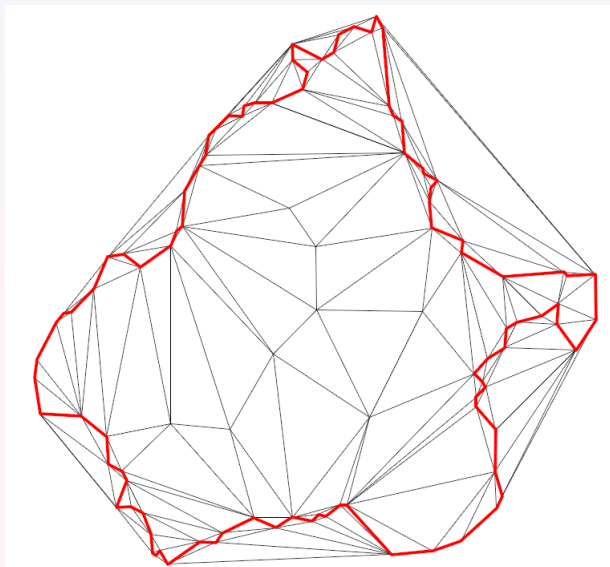
Oblast obsahuje podoblasti, uvnitř kterých nebude prováděna triangulace.

Otvory popsány v opačném pořadí, než nadřazená oblast.

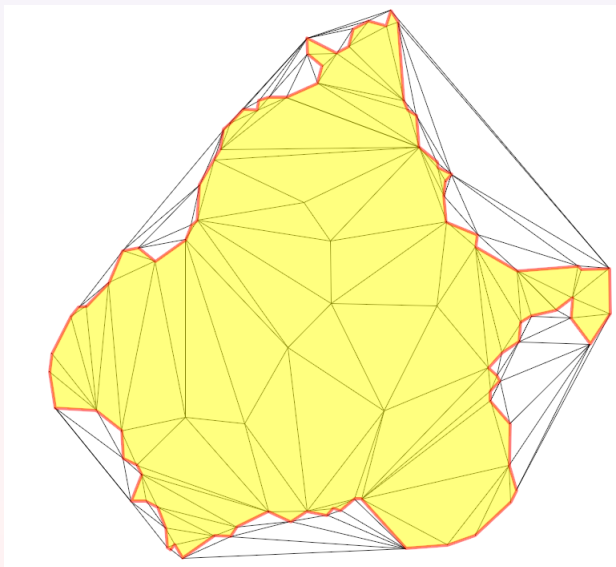
Lze rozlišit modifikovaným algoritmem pro PLP.

Použití při tvorbě DMT, místa bez vrstevnic.

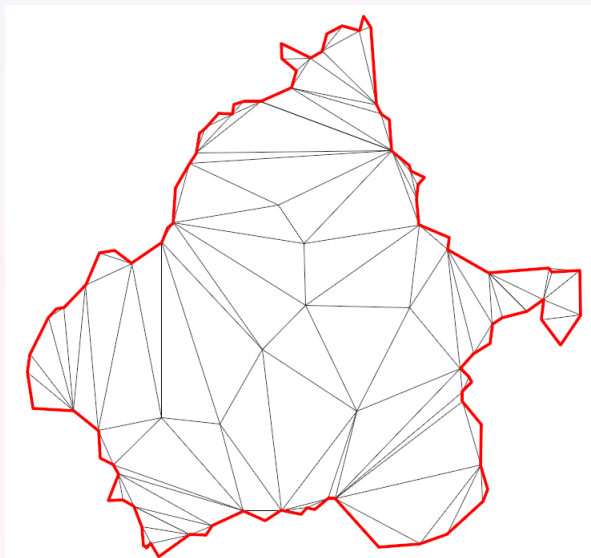
Typické pro vodní plochy, stavební objekty, místa s příliš velkým spádem...

71. Triangulace $DT(P)$ 

72. Selekce trojúhelníků uvnitř oblasti



73. Odstranění trojúhelníků vně oblasti



74. Datově závislé triangulace

U všech výše uvedených 2D triangulací tvar trojúhelníkové sítě ovlivňuje pouze poloha bodu, souřadnice z nehraje roli.

Takové triangulace nejsou bez dodatečných informací o terénu (kosterní čáry) vhodné k jeho modelování.

Data Dependent Triangulation (DDT) tyto nedostatky odstraňuje.

Vznikají optimalizací vstupní triangulace (nejčastěji DT) s využitím heuristik či genetických algoritmů.

Výhody:

- DDT berou v potaz výšku bodu, snaha o optimalizaci tvaru trojúhelníkové sítě.
- Trojúhelníkový model lépe zohledňuje skutečný tvar terénu.
- Automatická detekce terénních zlomů, netřeba zadávat povinné hrany.

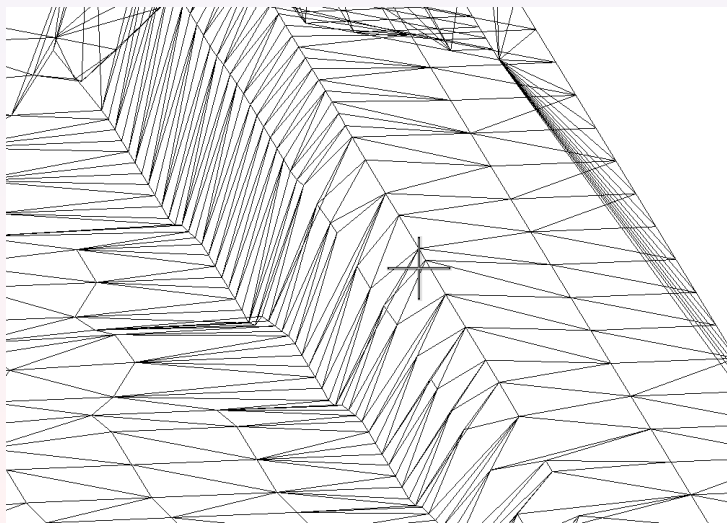
Nevýhody:

- Optimalizace heuristikou rychlá, zlepšení většinou nebývá významné.
- Optimalizace genetickými algoritmy kvalitní, avšak výpočetně náročné, vhodné pouze pro malé množiny ($n < 50000$).

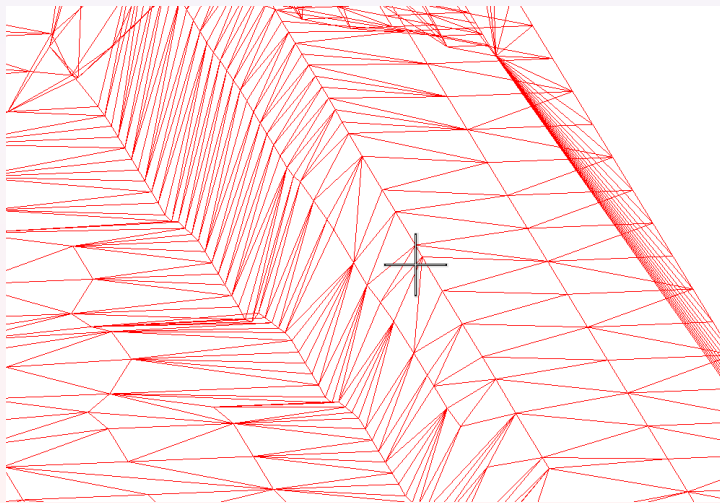
Dvě metody optimalizace:

- lokální optimalizace,
- modifikovaná lokální optimalizace.

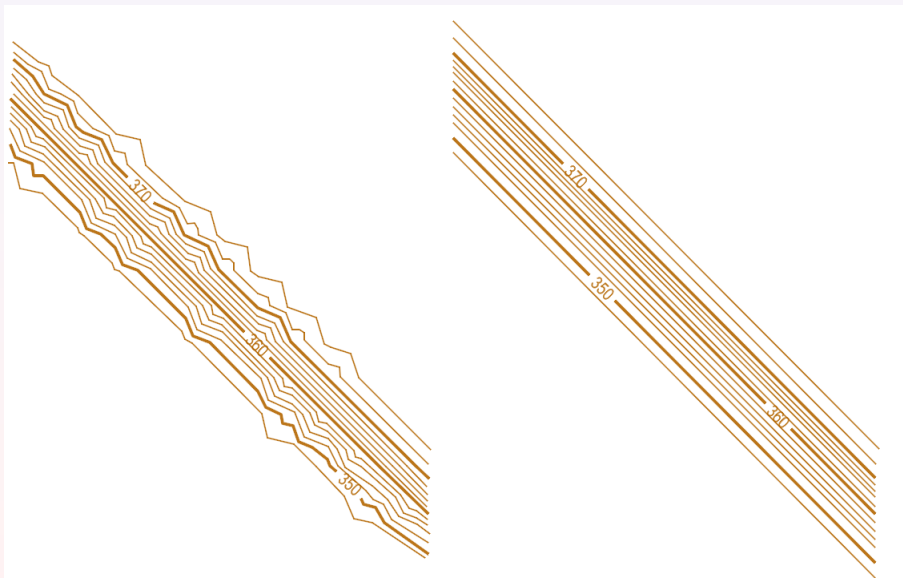
75. DT, nevhodné vystižení terénní hrany



76. DDT, lepší vystižení terénní hrany



77. Srovnání vrstevnic DT a DDT



78. Lokální optimalizace triangulace

Optimalizace triangulace vzhledem zadanému kritériu.
Realizována lokálním prohazování hran.

Používána greedy strategie.
Snaha dosáhnout globálního minima opakovaným hledáním lokálního minima.

V jednom kroku optimalizována “malá” část triangulace:

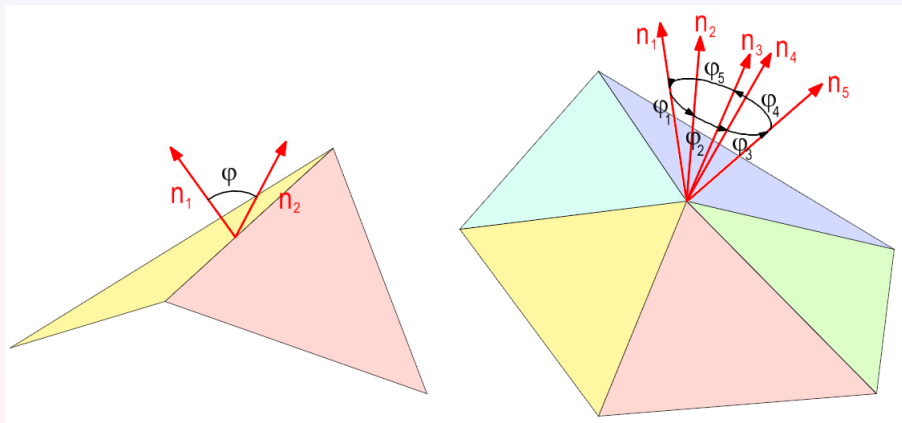
- *Edge Based Optimization*
Prováděna vzhledem ke každé hraně sdílené dvojicí trojúhelníků.
Častější varianta.
- *Vertex Based Optimization*
Prováděna vzhledem ke každému vrcholu sdíleného trojúhelníky.

Rychlé, avšak nepřilíš významné zlepšení.

Možné uvíznutí v lokálním minimum, nepřipouští dočasné zhoršení stavu.

Vysoká rychlost konstrukce.

79. Edge Based vs Vertex Based Optimization



80. Edge Based Optimization

Každé hraně e_i trojúhelníka \triangle přiřadí funkce c ohodnocení c_i

$$c_i = c(e_i).$$

Ohodnocovací funkce měří “ostrost přechodu” mezi trojúhelníky (např. úhel normál).
Cílem co nejhladší povrch polyedru.

Optimální triangulace minimalizuje globální kritérium

$$C(\tau) = \sum_{i=1}^m c_i = \min .$$

Globální kritérium neumíme exaktně minimalizovat (NP), použití heuristik.
Greedy strategie minimalizující

$$C(\tau(k)) = C(\tau(k-1)) + \Delta c(e),$$

minimalizací ceny updatu.
Konvergence pouze k lokálnímu minimu.

81. Lokální swap kritérium

Výchozí triangulace $\mathcal{T}(P)$: Hrana e_i , inciduje s $\triangle t_1, t_2$.

2 varianty:

1) Hrana e_i sdílená 2 trojúhelníky

Lokální swapovací kritérium: $\mathcal{T}(P)$ je optimální vzhledem k c právě když

$$c(e_i) < c(e'_i).$$

Nízká účinnost, malé území.

2) Hrana e_i sdílená 2 trojúhelníky + incidující hrany.

Celkem 6 trojúhelníků, větší účinnost.

Sousedící hrany v t_1 : e_j^1, e_j^2 , sousedící hrany v t_2 : e_j^3, e_j^4 , ohodnocení

$$C(\tau) = c(e_i) + \sum_{k=1}^4 c(e_j^k)$$

Swap \Rightarrow triangulace $\mathcal{T}'(P)$ s trojúhelníky t'_1 a t'_2 , ohodnocení

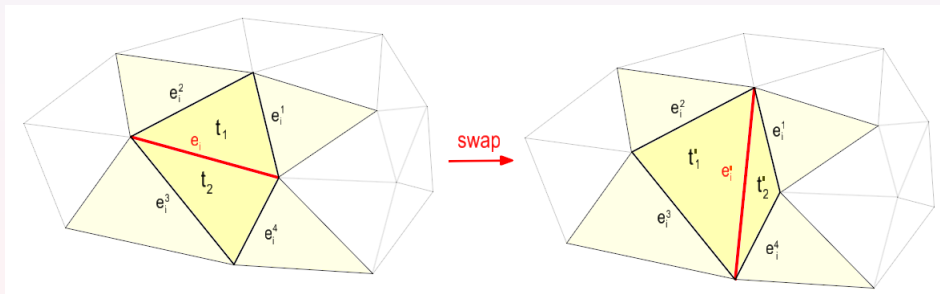
$$C(\tau') = c(e'_i) + \sum_{k=1}^4 c(e_j^k).$$

Lokální swapovací kritérium: $\mathcal{T}(P)$ je optimální vzhledem k c právě když

$$C(\tau) < C(\tau') \Leftrightarrow c(e_i) + \sum_{k=1}^4 c(e_j^k) < c(e'_i) + \sum_{k=1}^4 c(e_j^k).$$

Výběr triangulace s nižší hodnotou C .

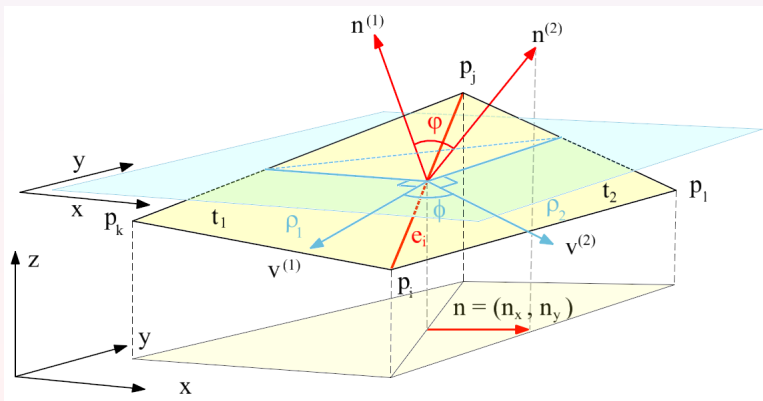
82. Ukázka lokální optimalizace



83. Lokální kritéria pro DDT

Přehled kritérií:

- Angle Between Normals (ABN).
- Distance From Planes (DFP).
- Smoothnes of Contours (SCO).



84. Přehled kritérií

Rovina ϱ_i

$$\varrho_i(x, y) = a_i x + b_i y + c_i z$$

Angle Between Normals:

Úhel φ mezi normálami $n^{(1)}, n^{(2)}$ trojúhelníků t_1, t_2

$$c_{ABN}(e_i) = \varphi = \arccos \frac{n^{(1)} \cdot n^{(2)}}{\|n^{(1)}\| \|n^{(2)}\|}, \text{ kde } n^{(i)} = \left(\frac{\partial \rho_i}{\partial x_i}, \frac{\partial \rho_i}{\partial y_i}, \frac{\partial \rho_i}{\partial z_i} \right) = (-a_i, -b_i, 1)$$

Distance From Planes:

Součet vzdálenost bodů p_k, p_l od rovin ϱ_1, ϱ_2 .

$$c_{DFP}(e_i) = (d(p_k, \varrho_1)^p + d(p_l, \varrho_2)^p)^{1/p}, \text{ kde } p = 1, 2.$$

Smoothness Of Contours:

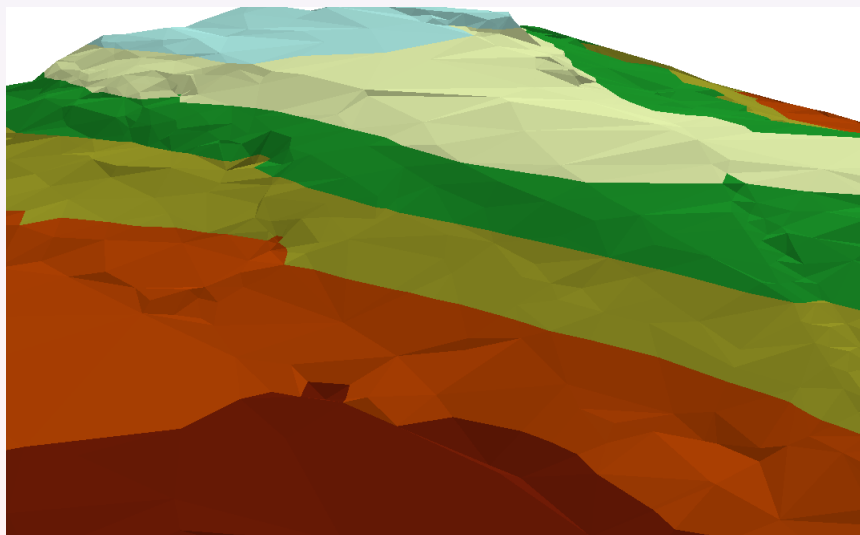
Úhel ϕ mezi $v^{(1)}, v^{(2)}$ (průměty $n^{(1)}, n^{(2)}$ v ϱ_0 , vodorovná rovina)

$$c_{SCO}(e_i) = \phi = \arccos \frac{v^{(1)} \cdot v^{(2)}}{\|v^{(1)}\| \|v^{(2)}\|} = \cos^{-1} \frac{a_1 a_2 + b_1 b_2}{\sqrt{a_1^2 + b_1^2} \sqrt{a_2^2 + b_2^2}}.$$

85. Znáznornění DMT: trojúhelníková síť



86. Znáznornění DMT: barevná hypsometrie (kvantitativní použití barev)



87. Polyedrický model terénu

Plošky jsou představovány trojúhelníky se společnou hranou.
Síť trojúhelníků vytvořena za použití triangulačních algoritmů.

Proložením rovin vrcholy jednotlivých trojúhelníků v \mathbb{R}^3 .
Vznikne nepravidelný mnohostěn, tzv. **polyedr**, který se přimyká k terénu.

V trojúhelnících pouze lineární interpolace, což pro řadu účelů nepostačuje.

Do polyedrického modelu lze zadat **povinné hrany**.

Nejčastěji hřbetnice, údolnice, spádnice.

Zlepšení jeho aproximačních vlastností.

Vkládání povinných hran: Constrained Delaunay Triangulation.

Nepravidelné rozložení bodů, pro interpolaci používáme specializované metody.

Nejčastěji: IDW, Krigging.

88. Konstrukce polyedrického modelu

Vrcholy $p_1 = [x_1, y_1, z_1]$, $p_2 = [x_2, y_2, z_2]$, $p_3 = [x_3, y_3, z_3]$ každého trojúhelníku t proložíme rovinu \mathcal{T}

$$z = ax + by + c.$$

Koeficienty a , b , c představující složky normálového vektoru roviny

$$a = \frac{\begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad b = \frac{\begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad c = \frac{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}.$$

Rovnice jednotlivých rovin nejsou udržovány v paměti.

Podle potřeby operativně určovány (on the fly) \Rightarrow výhoda pro práci s rozsáhlými modely.

89. Interpolace vrstevnic

Lineární interpolační algoritmy:

Lineární spád terénu mezi interpolovanými body.
Rozestup vrstevnic mezi dvěma body je konstantní.
Výpočetně jednoduché, ale nevystihuje realitu.

Nelineární interpolační algoritmy:

Mezi interpolovanými body předpokládáme plynulou změnu sklonu terénu.
tzv. geomorfologická interpolace, mapy středních měřítek.

Rozestup vrstevnic mezi dvěma body není konstantní.
Zohledňuje skutečný tvar terénu (sklon okolních plošek).
Využití kvadratické či kubické interpolace, plátování.

Používá se v mapách velkých a středních měřítek.
Postup je značně složitý a obtížně se algoritmizuje.

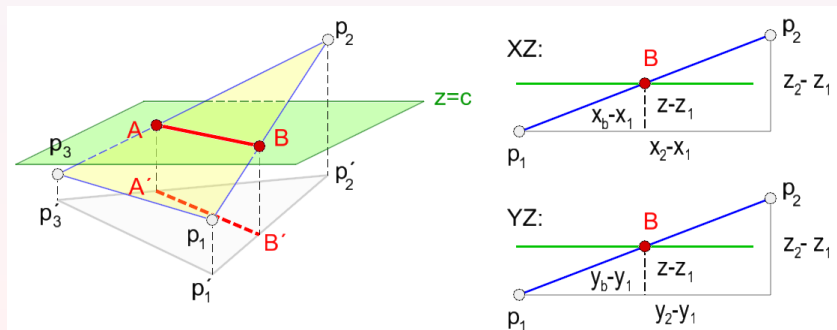
90. Konstrukce vrstevnic lineární interpolací

Dáno: Rovina plátu $\mathcal{T}(p_1, p_2, p_3)$, rovina $\rho : z = h$.

Hledáme: Průsečnice \overline{AB} rovin ρ a \mathcal{T} .

Založena na analytické geometrii: hledání průsečnice roviny \mathcal{T} určené trojúhelníkem $t \in \mathcal{DT}$ a vodorovné roviny s výškou h .

Opakováno nad všemi t .



91. Výpočet souřadnic bodů A, B

Varianty vzájemné polohy ρ a τ :

- Nemají žádný společný bod (neřešíme).
- Průsečnice tvoří 1 bod (neřešíme).
- Průsečnice je úsečka.

Z podobnosti trojúhelníků představujících průměty do roviny XZ a YZ platí:

$$\frac{x_2 - x_1}{z_2 - z_1} = \frac{x_b - x_1}{z - z_1}, \quad \frac{y_2 - y_1}{z_2 - z_1} = \frac{y_b - y_1}{z - z_1},$$

$$\frac{x_3 - x_1}{z_3 - z_1} = \frac{x_a - x_1}{z - z_1}, \quad \frac{y_2 - y_1}{z_2 - z_1} = \frac{y_a - y_1}{z - z_1}.$$

Výsledné souřadnice koncových bodů A, B průsečnice určíme ze vztahů

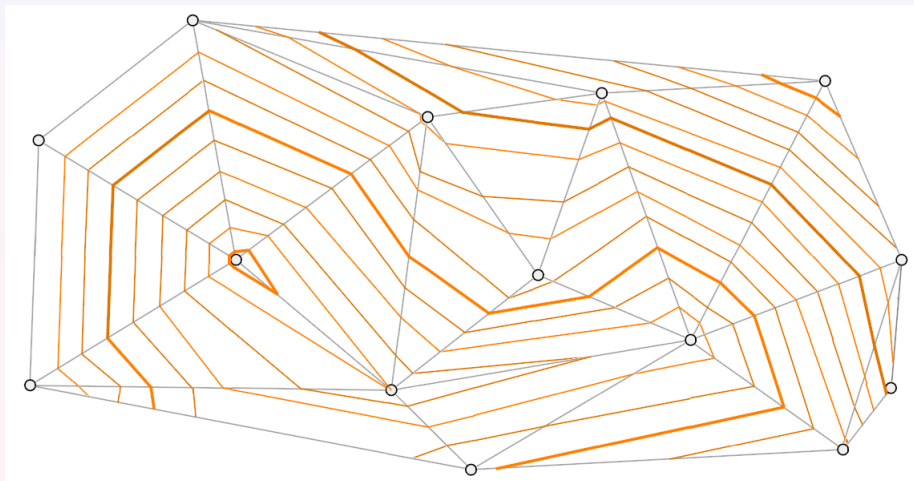
$$x_a = \frac{x_3 - x_1}{z_3 - z_1}(z - z_1) + x_1, \quad x_b = \frac{x_2 - x_1}{z_2 - z_1}(z - z_1) + x_1,$$

$$y_a = \frac{y_3 - y_1}{z_3 - z_1}(z - z_1) + y_1, \quad y_b = \frac{y_2 - y_1}{z_2 - z_1}(z - z_1) + y_1.$$

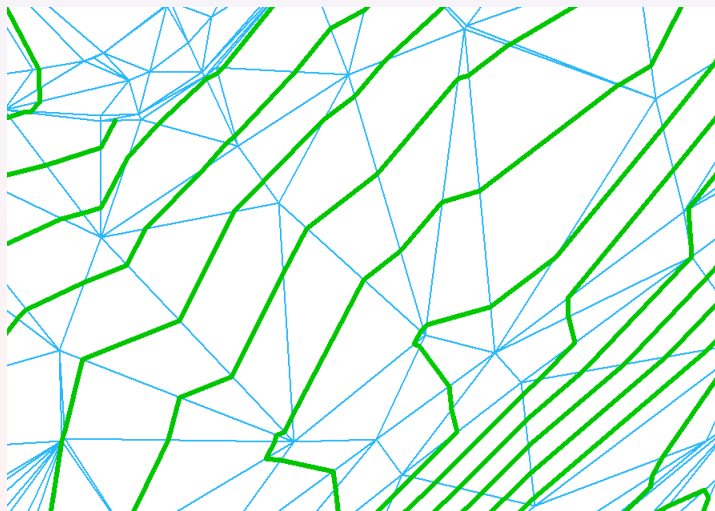
Test, zda rovina ρ protíná stranu (p_i, p_{i+1}) trojúhelníku:

$$(z - z_i)(z - z_{i+1}) < 0.$$

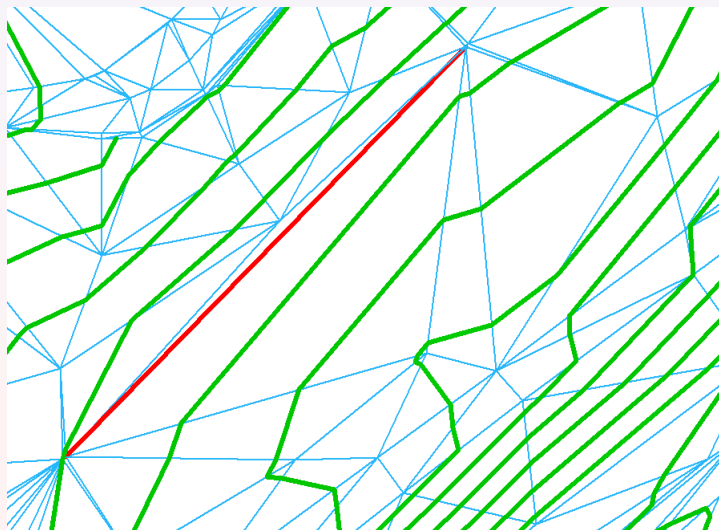
92. Ukázka výpočtu vrstevnic lineární interpolací (DT)



93. Vliv vložení povinné hrany do triangulace: výchozí situace



94. Vliv vložení povinné hrany do triangulace: lokální změna průběhu DMT



95. Analýza sklonu terénu

Analytická úloha realizovaná nad DMT.

Zprostředkující hodnotou je *gradient* (tj. vektor max. spádu).

Gradient $\nabla f(x_0, y_0, z_0)$ funkce $f(x, y, z)$ v bodě $p = [x_0, y_0, z_0]$

$$\nabla f(x_0, y_0, z_0) = \left(\frac{\partial f}{\partial x}(x_0), \frac{\partial f}{\partial y}(y_0), \frac{\partial f}{\partial z}(z_0) \right).$$

Rovnice roviny ρ

$$\rho : ax + by + cz + d = 0.$$

Gradient $\nabla \rho(x_0, y_0, z_0)$ roviny ρ

$$\nabla \rho(x_0, y_0, z_0) = \left(\frac{\partial \rho}{\partial x}(x_0), \frac{\partial \rho}{\partial y}(y_0), \frac{\partial \rho}{\partial z}(z_0) \right) = (a, b, c).$$

Použití pro analýzu hydrologických poměrů, sesuvů, lavin, návrhy komunikací, stavebních objektů.

96. Analýza sklonu terénu

Rovina ρ procházející 3 body

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0.$$

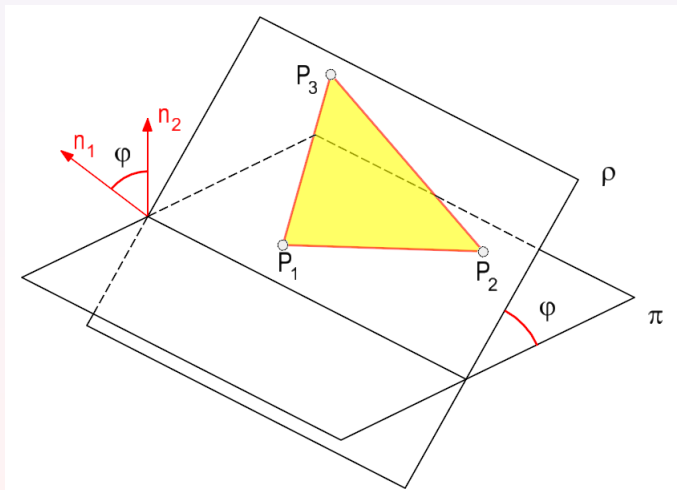
Odchylka φ rovin ρ a π :

$$\begin{aligned} \varphi &= \arccos \frac{n_1 \cdot n_2}{\|n_1\| \|n_2\|} = \arccos \frac{c}{\|n_1\|}, \\ n_1 &= (a, b, c), \\ n_2 &= (0, 0, 1). \end{aligned}$$

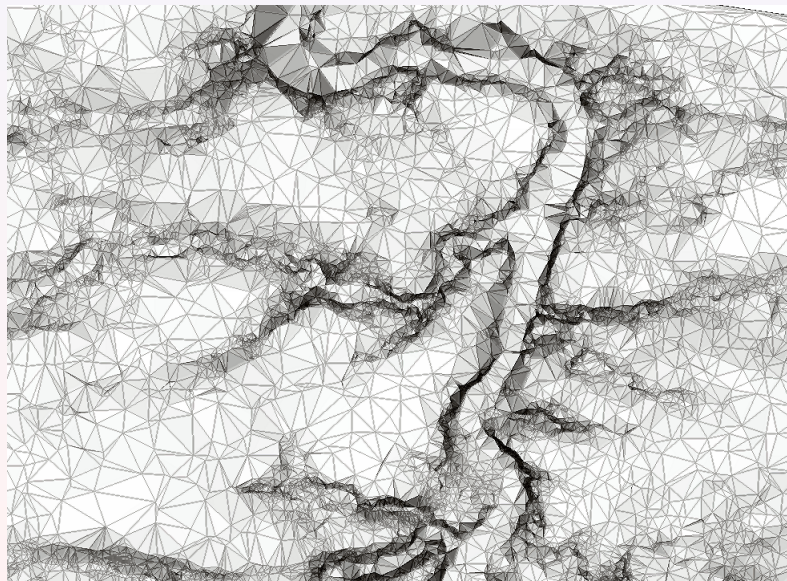
Výpočet prováděn nad každým trojúhelníkem t DMT.

Vizualizace trojúhelníků (tj. vyplnění barvou) na základě hodnoty φ .

97. Sklon terénu



98. Vizualizace sklonu terénu



99. Analýzy orientace terénu

Definována jako azimut průmětu gradientu $\nabla\rho$ do roviny x, y .

Vektor v průmětem gradientu $\nabla\rho(x_0, y_0, z_0)$ do roviny xy

$$v = \left(\frac{\partial\rho}{\partial x}(x_0), \frac{\partial\rho}{\partial y}(y_0), 0 \right) = (a, b, 0).$$

Azimut A vektoru v měřen od osy y

$$A = \arctan\left(\frac{a}{b}\right).$$

Výpočet prováděn nad každým trojúhelníkem DMT.

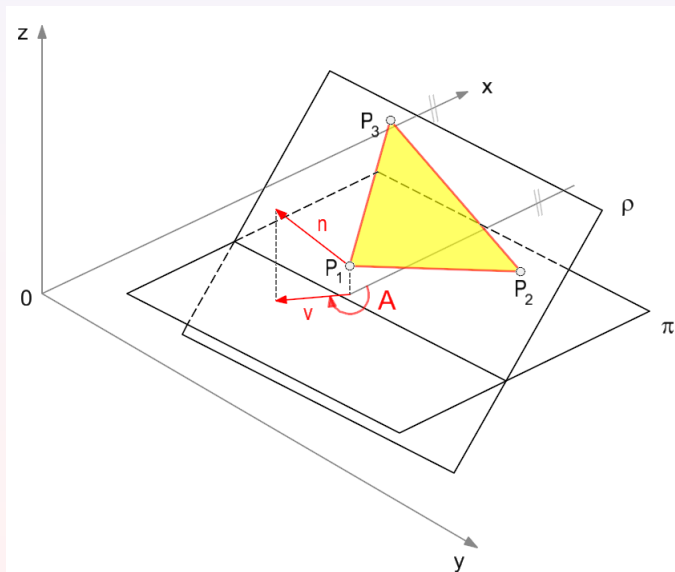
Pozor na správnou detekci kvadrantů.

Využití ve stavebnictví, zemědělství, hydrologii.

Sluneční svit ovlivňuje množství tepla dopadajícího na zemský povrch.

Důležité pro hydrologické poměry, růst zemědělských plodin.

100. Orientace terénu



101. Vizualizace orientace terénu

