Rovinné triangulace a jejich aplikace

I.Kolingerová

Obsah

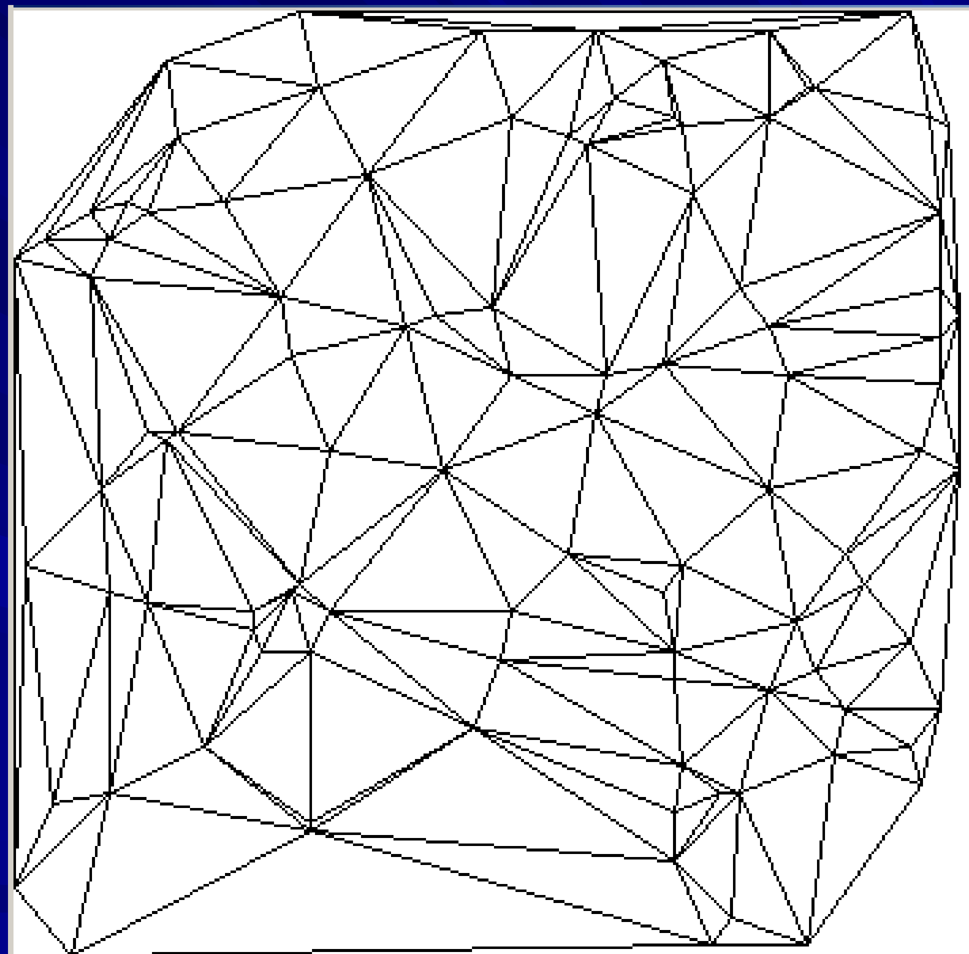
A) Definice triangulace

B) Triangulace v 2D

1. Kritéria kvality sítě
2. Hlavní typy
3. Delaunayova triangulace
4. Greedy triangulace
5. Triangulace s minimální váhou
6. Triangulace s omezením
7. Datově závislé triangulace
8. Multikriteriálně optimalizovaná triangulace
9. Aplikace

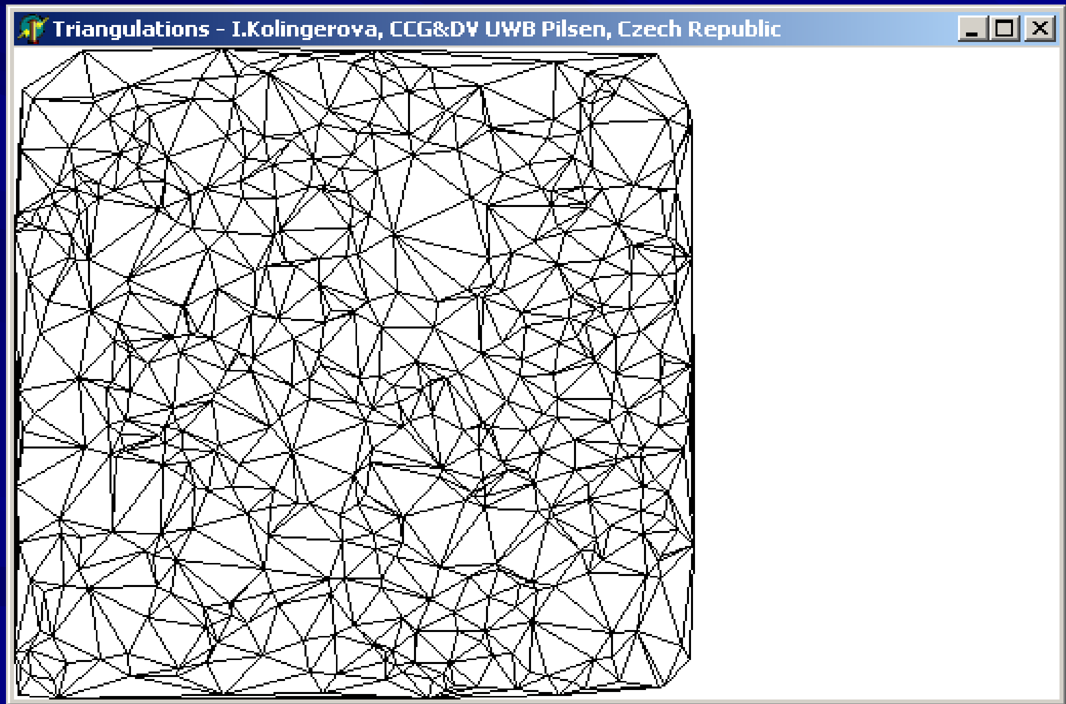
A) Definice triangulace

Triangulace *T(P)* množiny *N* bodů v E^d - rozdělení *CH(P)* na simplexy s vrcholy z *P*



- složitost nejméně $O(N \log N)$ v E^2 a $O(N^2)$ v E^3

- Počet hran v E^2:
  - nejvýše $3N-6$
  - přesně $3N-3-N_{CH}$ v E^2



B) Triangulace v 2D

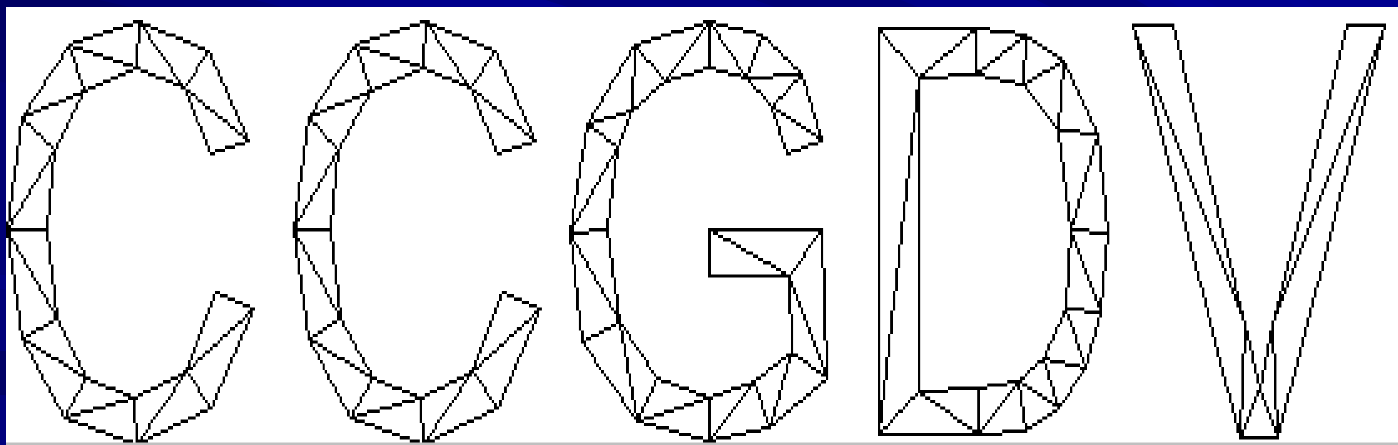
1. Kritéria kvality sítě

- pro danou množinu bodů mnoho triangulací = > kterou vybrat ?

- „co nejrovnostrannější trojúhelníky“
  = > úhlová kritéria (maximalizace min. úhlů, minimalizace maximálních úhlů) a hranová kritéria (co nejkratší součet délek hran)

Další požadavky:

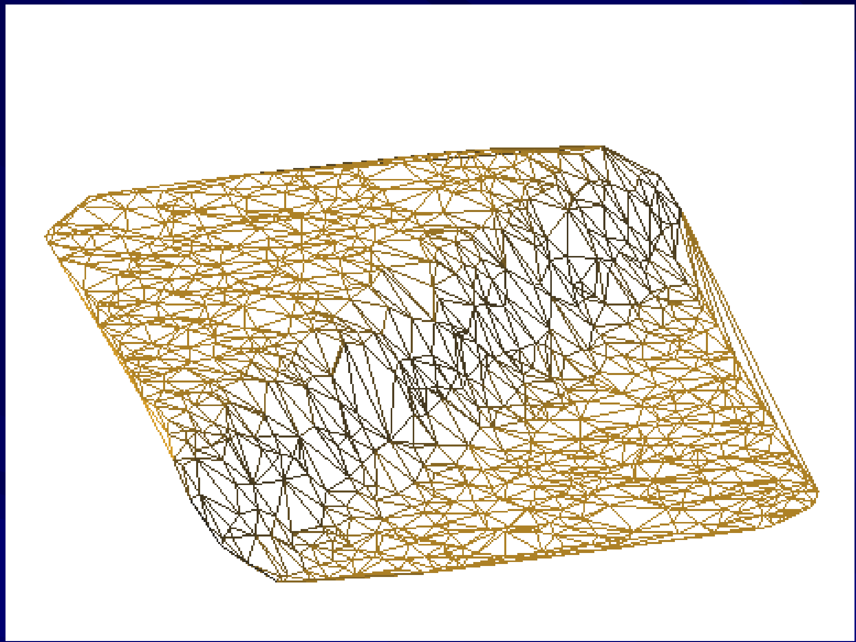
- včlenění určitých povinných hran (constraints = > triangulace s omezením - constrained triangulation)
- zachování nekonvexního tvaru oblasti ...

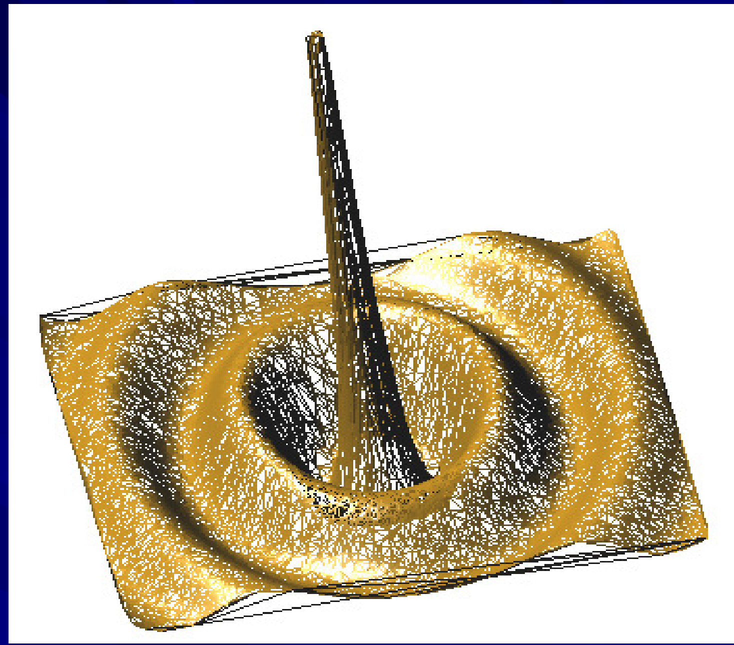


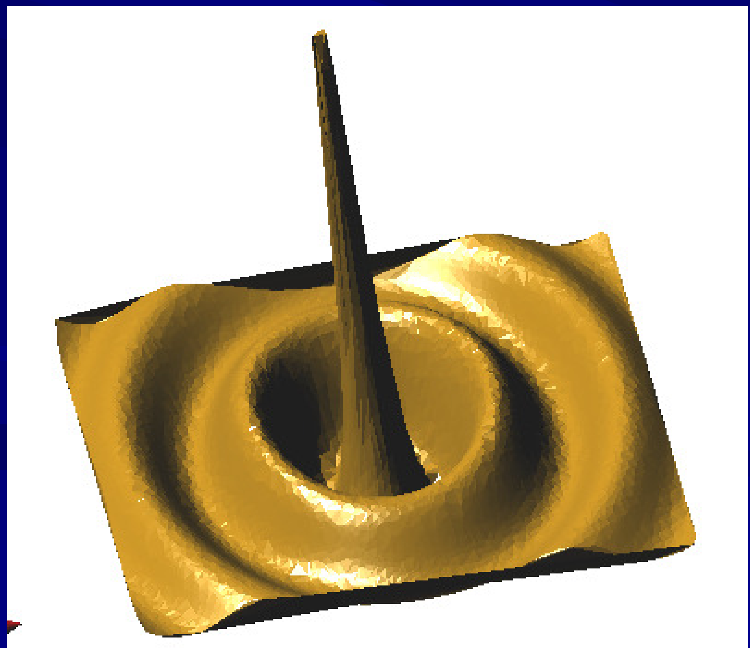
Požadavky na algoritmus:

- nízká složitost v nejhorším i očekávaném případě
- snadnost implementace
- numerická robustnost, malá citlivost na singulární případy
- použitelnost pro vyšší dimenzi
- použitelnost pro přibývání a ubývání bodů
- paralelizovatelnost

FEM varianta: dána pouze hranice oblasti







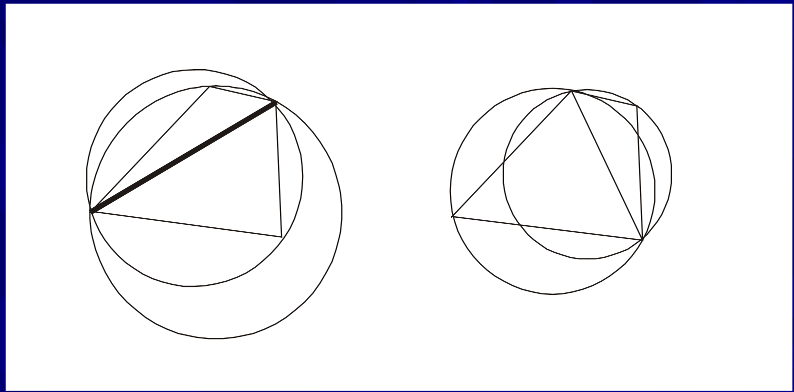
Příklady triangulací

2. Hlavní typy triangulací

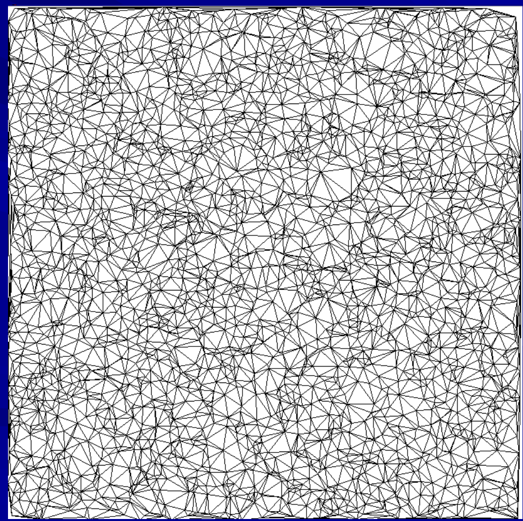
- Delaunayova - *DT*
- žravá (hltavá, greedy) - *GT*
- triangulace s minimální váhou - *MWT*
- triangulace s omezením (constrained triangulation) - *CDT, CGT*
- datově závislé triangulace – *DDT*
- multikriterálně optimalizované triangulace

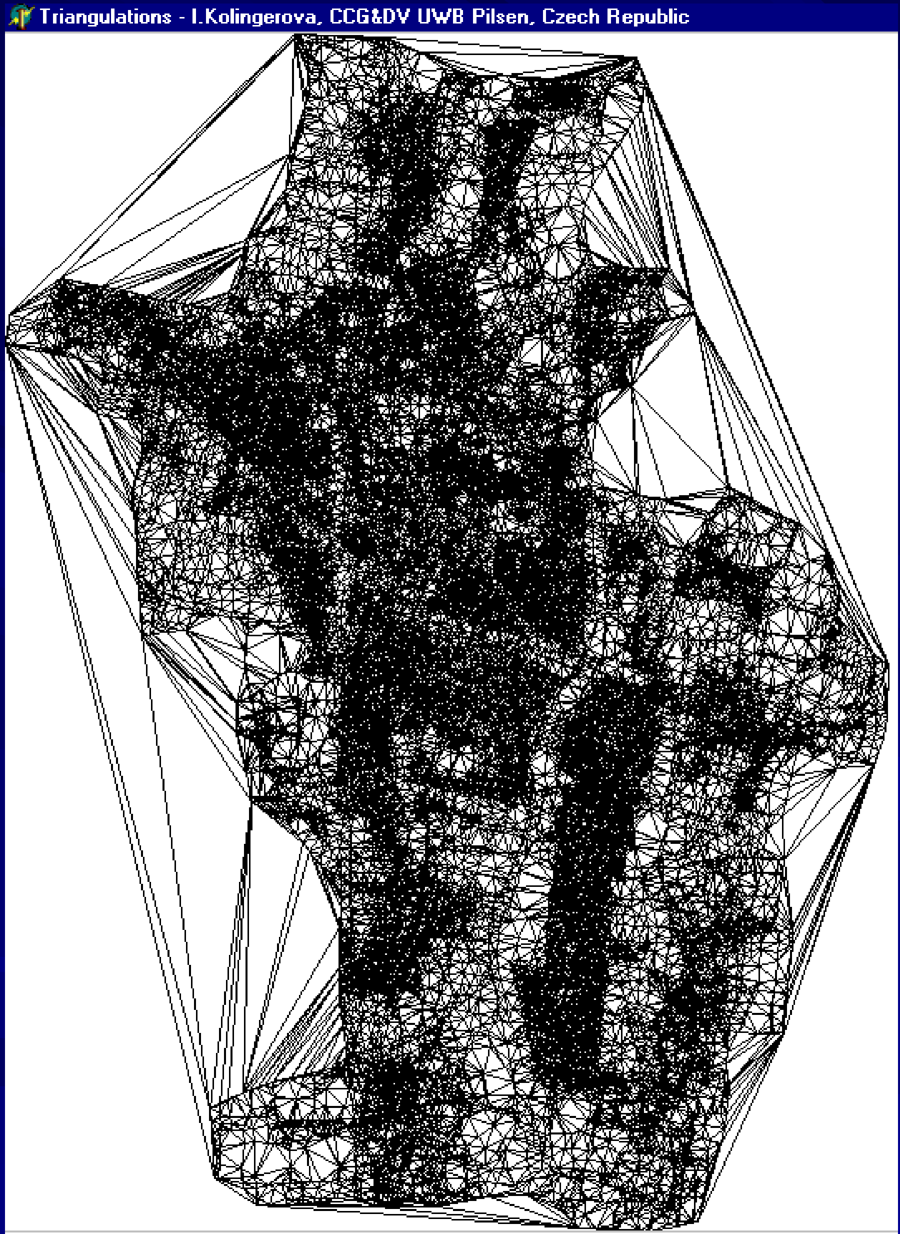
3. Delaunayova triangulace

- kružnice opsaná libovolnému trojúhelníku z *DT(P)* neobsahuje ve svém vnitřku žádný bod z *P*



- maximalizuje minimální úhel
- ze všech triangulací trojúhelníky nejblíže k rovnostranným



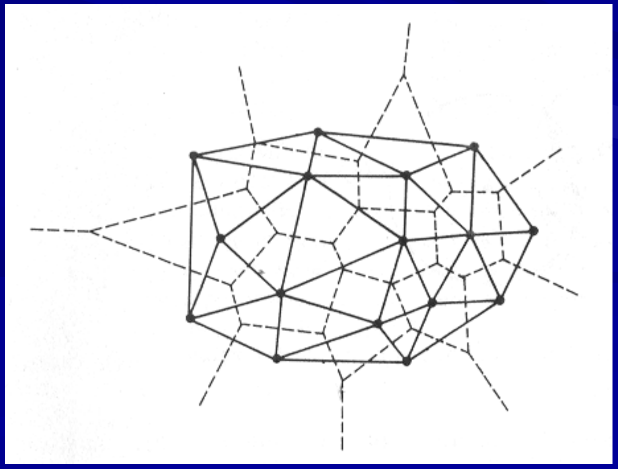


Reálná datová množina
(60 244 bodů,
120 465 trojúhelníků)

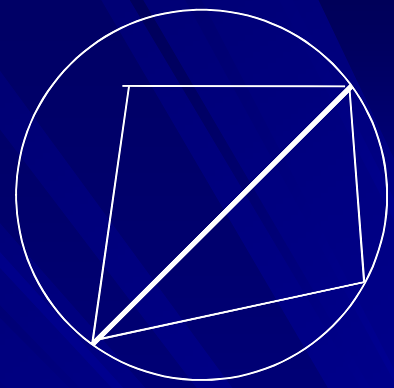
Hlavní algoritmy výpočtu *DT*:

- přímé
  - lokální prohazování hran
  - inkrementální vkládání
  - inkrementální konstrukce
  - převod do vyšší dimenze
  - rozděl a panuj
  - zametání ...
- nepřímé - přes Voronoiův diagram

(Etalon: Shewchuk:Triangle)



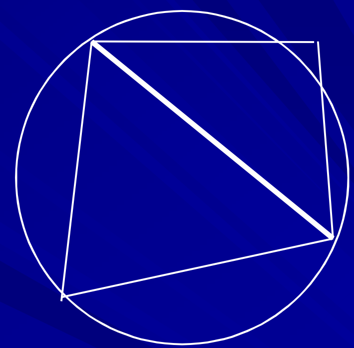
Lokální prohazování



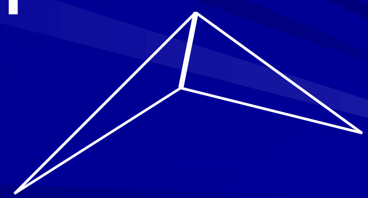
Vstup: ***P* – množina bodů v rovině**

Výstup: ***LOT(P)* – Lokálně optimální triangulace**

1. **begin**
2. Zkonstruuj počáteční triangulaci *T(S);*
3. **while** *T(S)* není lokálně optimální **do**



4. **begin**
5. **If** existuje vnitřní hrana *e*, kt. není lokálně optimální **then**



6. flip hrany *e*;
7. **end;**
9. **end**

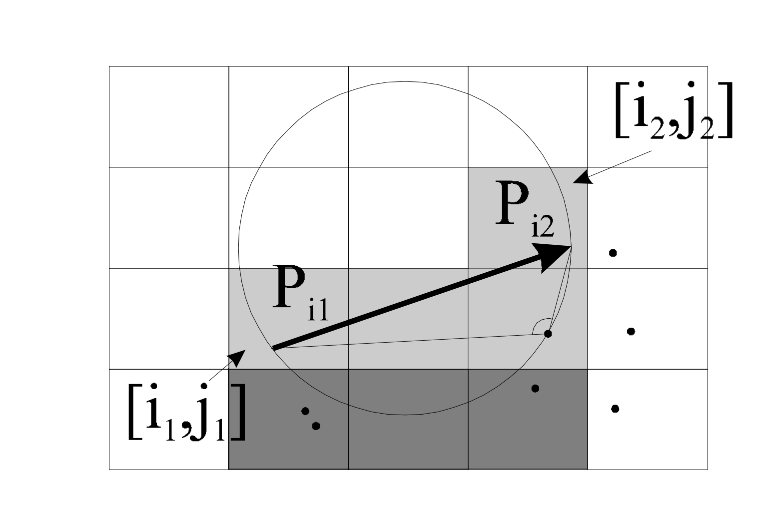
$O(N^2)$, resp. $O(N)$

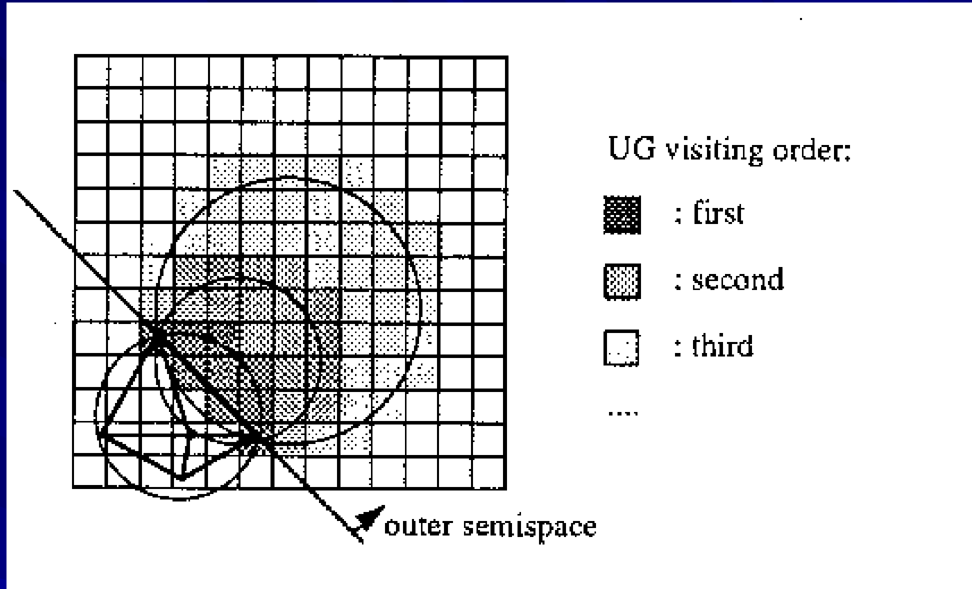
Lokální prohazování

Vlastnosti:

- velmi jednoduché,
- oček.čas. složitost *O(N),*
- paměťové nároky *O(N)*
- není on-line, sériový,
- nemusí fungovat v 3D
- základ výpočtu pro libovolné lokální kritérium (např. pro *DDT*)

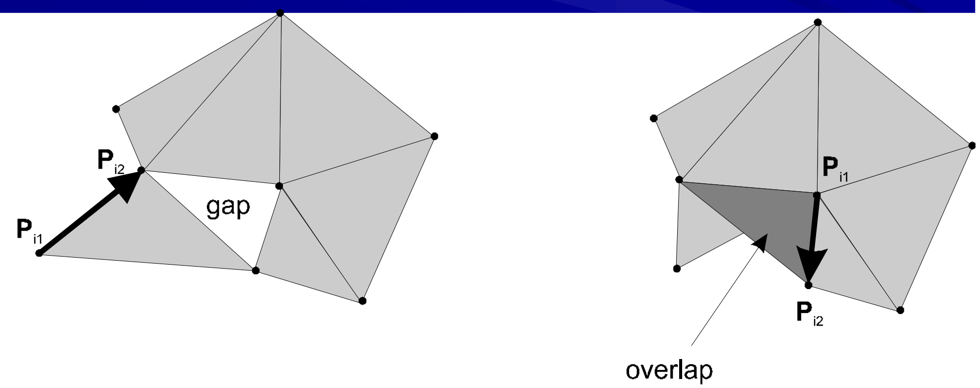
Inkrementální konstrukce



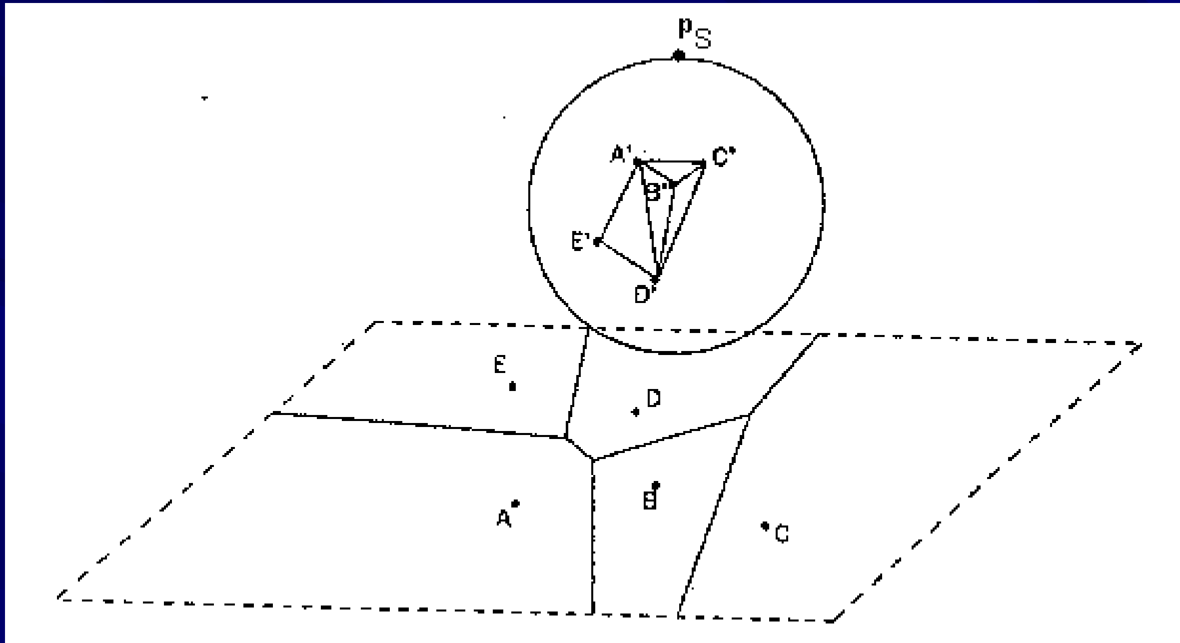


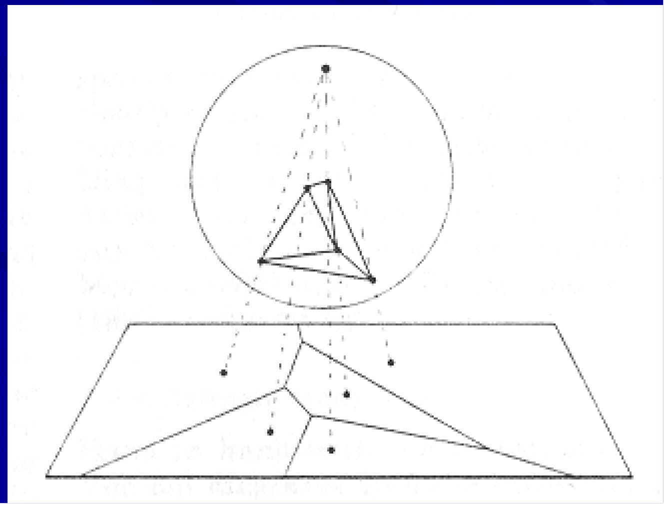
Vlastnosti:

- očekávaný čas *O(N log N)*, velmi pomalé,
- očekávaná paměť *O(N)*
- není on-line, sériový,
- rozšiřitelný do 3D,
- nestabilní



Včlenění do vyšší dimenze



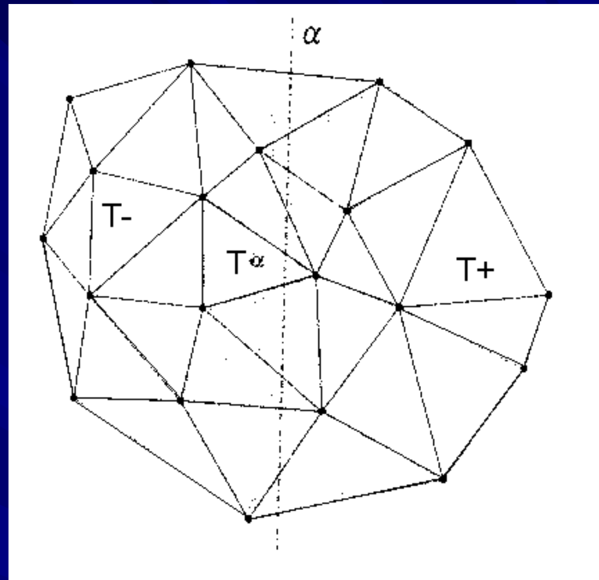


Vlastnosti:

dané algoritmem pro *CH(S)*

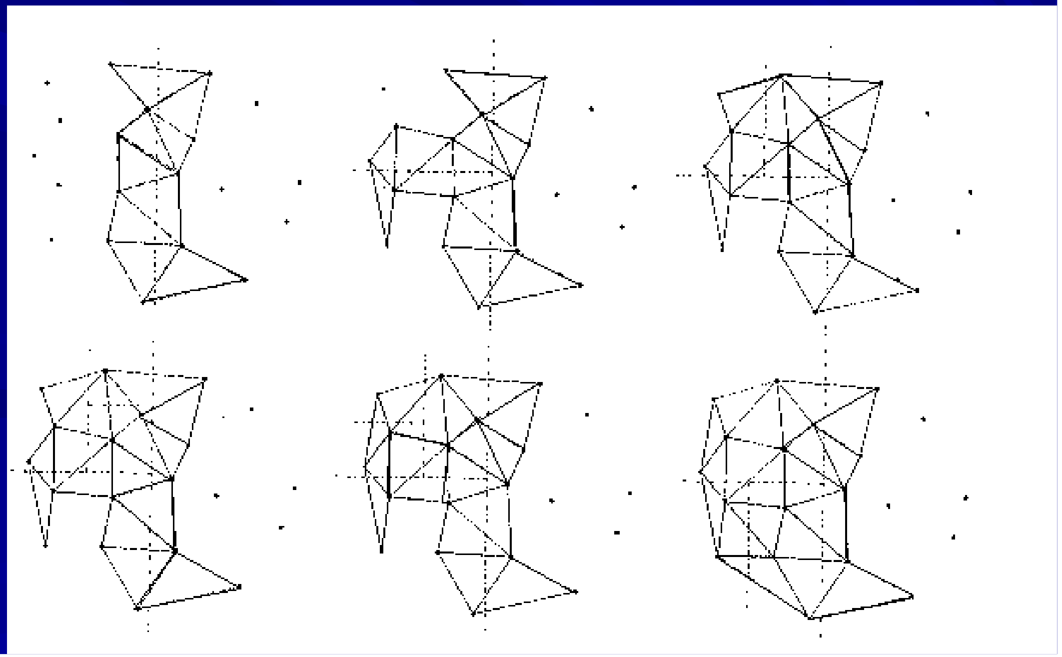
Rozděl a panuj

např. algoritmus De Wall



Dělení DT

do 3 skupin



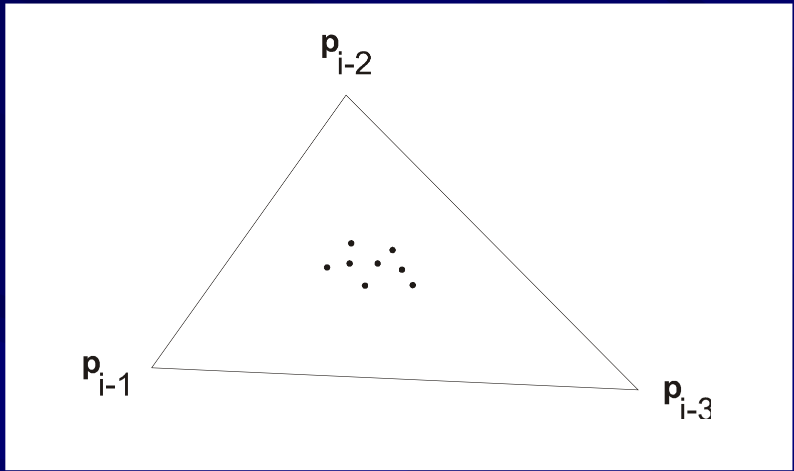
Některé fáze DeWall

Rozděl a panuj

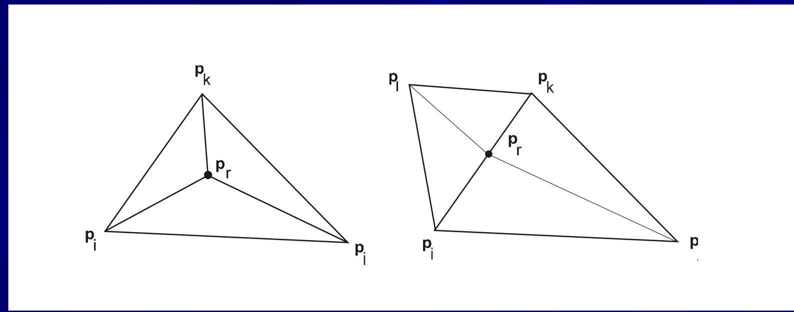
Vlastnosti:

- očekávaný čas *O(N),*
- očekávaná paměť *O(N),*
- není on-line,
- je rozšiřitelný do 3D ,
- náročná implementace,
- často paralelizován

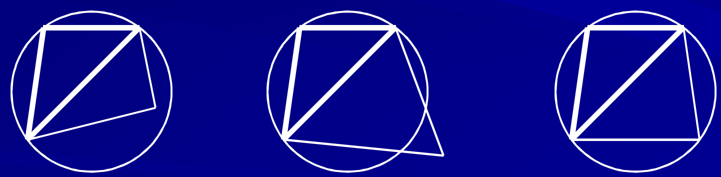
Inkrementální vkládání



Počáteční situace



Vkládání bodu



Swap (legalizace)

Inkrementální vkládání

Vstup: ***P*** **– množina bodů v rovině**

Výstup: ***DT(P)*** **– Delaunayova triangulace**

1. begin

2. Zkonstruuj počáteční trojúhelník přidáním 3 vnějších vrcholů;

3. Pro všechny body opakuj

4. begin

5. Najdi trojúhelník obsahující vkládaný bod;

6. Rozděl trojúhelník;

7. Rekurzivně legalizuj nové trojúhelníky;

8. end;

9. Odstraň všechny trojúhelníky obsahující přidané vrcholy;

10. end

$O(N^2)$, resp. $O(N \log N)$

Inkrementální vkládání

Lokace : hrubá síla

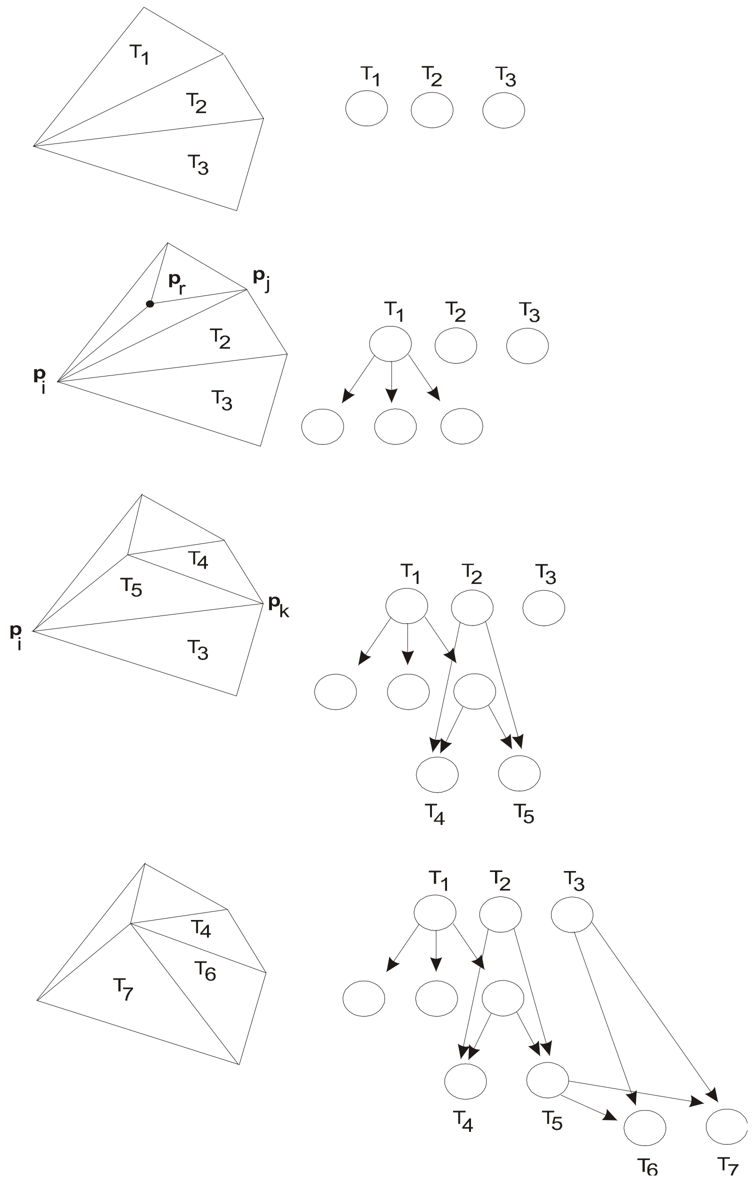
procházka

DAG ….

Vlastnosti (s DAG):

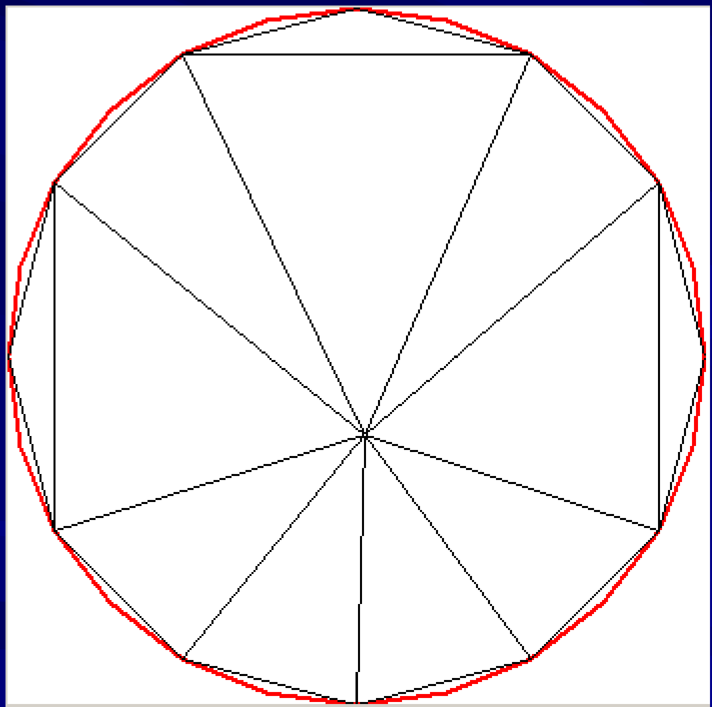
- oček. čas $O(N \log N)$,
- oček. paměť $O(N)$,
- sériový,

  ale lze paralelizovat
- rozšiřitelný do 3D

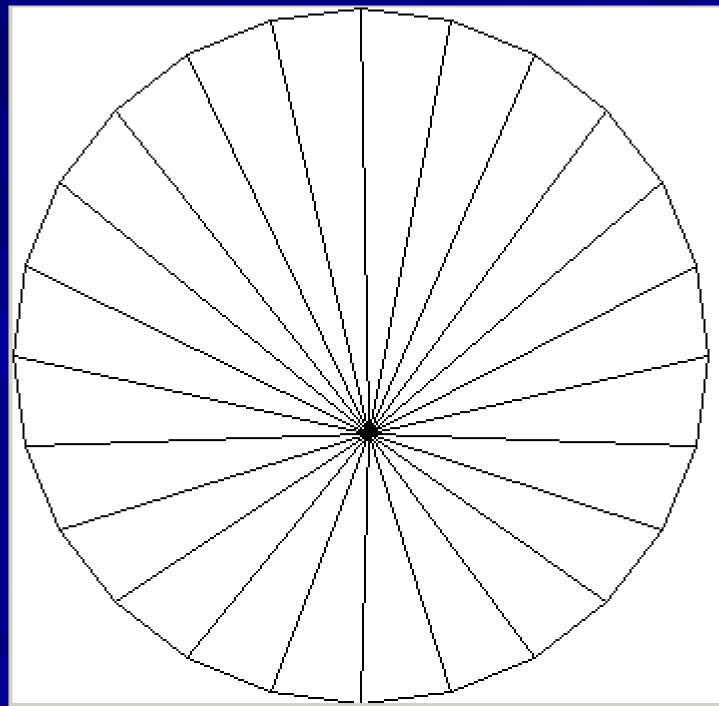


4. Žravá (greedy) triangulace

- obsahuje nejkratší vzájemně se neprotínající hrany
- složitost výpočtu $O(N^3)$ až $O(N)$



GT



DT

Výpočet *GT*: greedy algoritmus

Vstup: ***P* – množina bodů v rovině**

Výstup: ***GT(P)* – Žravá triangulace**

1. begin

2. Vytvoř všechny možné hrany a seřaď je podle délky vzestupně; $O(N^2 \log N)$

3. Přijmi nejkratší hranu do triangulace;

4. **Dokud** není triangulace hotova **opakuj** $O(N)$

5. **begin**

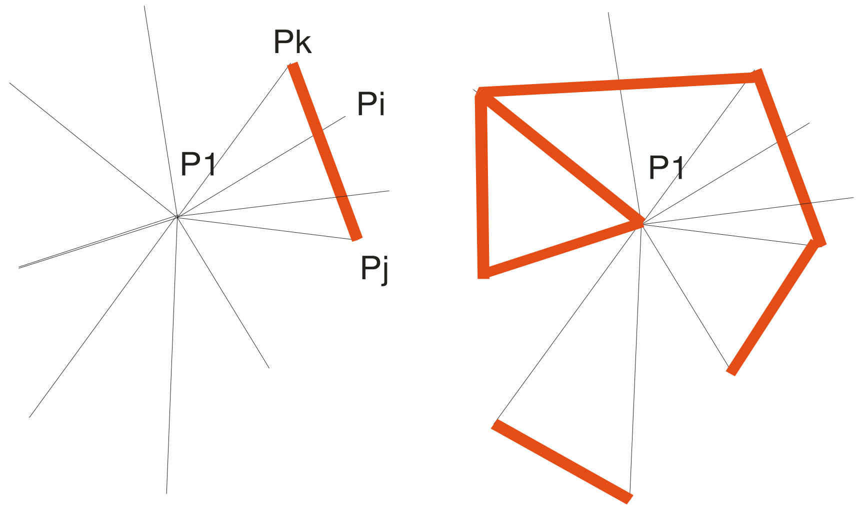
6. Vezmi další hranu v pořadí;

7. Pokud hrana neprotíná žádnou již přijatou hranu, přijmi ji do triangulace;

8. **end**

9. end $O(N^3)$, resp. $O(N^2 \log N)$

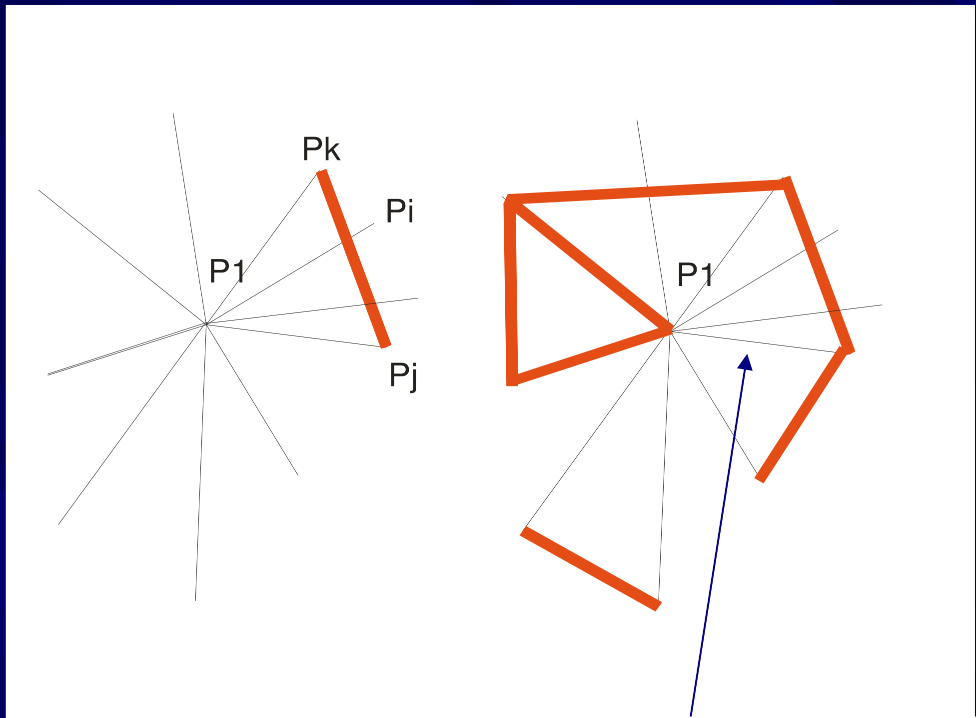
Jak docílit $O(N^2 \log N)$: Gilbert – test 1 hrany v $O(\log N)$



Příklad situace (červené hrany= už přijaté hrany triangulace)

- N-1 sektorů
- pro každý udržovat 1 hranu nejbližší do středu
- pro každý bod udržovat jednu hvězdu
- hranu testovat v hvězdě jejího počát. bodu
- přijatou hranu vložit do N-2 hvězd

Aktuální hrana p_1p_i - protíná nejbližší hranu k p_1 v daném sektoru? (Sektor $p_jp_1p_k$)



V oblasti uvnitř červených hranic (=už přijaté hrany triangulace) budou hrany přijaty

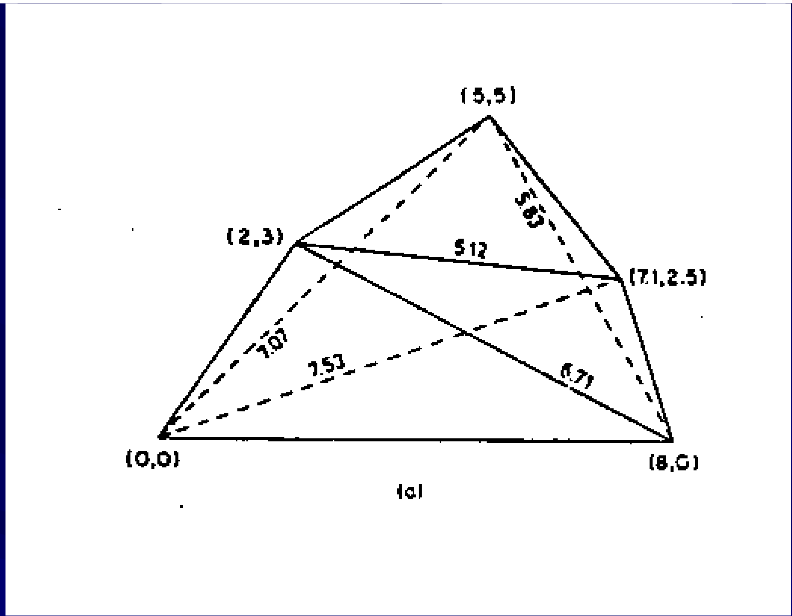
- Datová struktura: segmentový strom (sektory hvězdice jsou listy stromu, v uzlech jsou hrany nejbližší ke středu hvězdice)

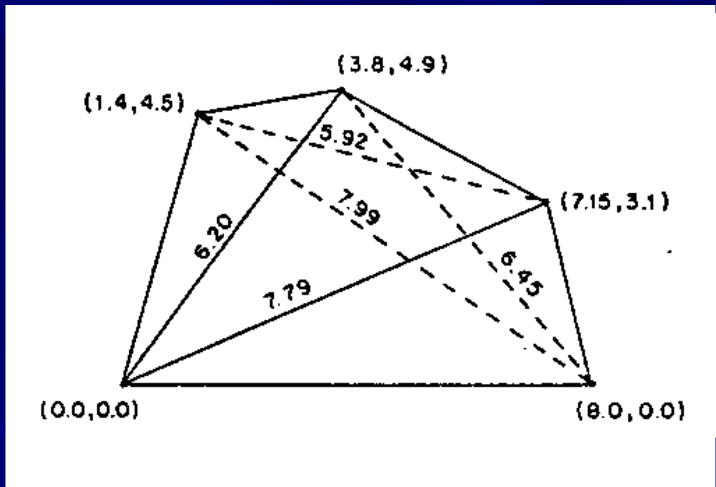
O(N² log N) čas, *O(N²)* paměť

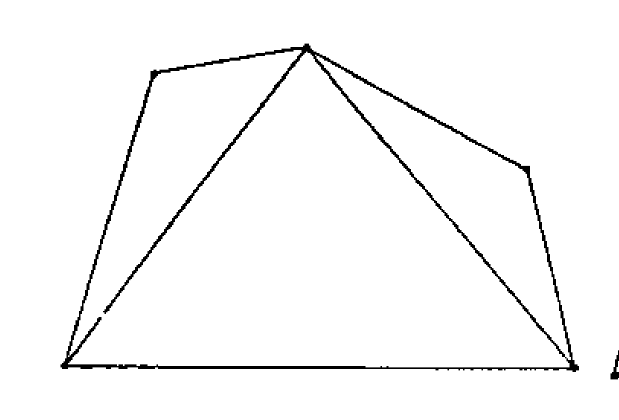
5. Triangulace s minimální váhou (Minimum weight triangulation, *MWT*)

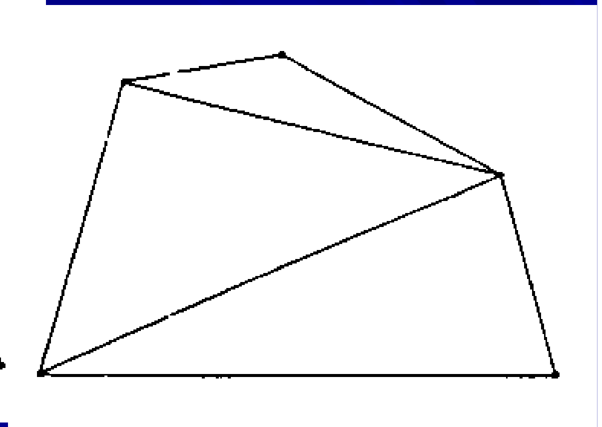
- váha = součet délek hran
- minimální váha ze všech triangulací *P* - globální extrém
- není známo, zda jde spočítat v polynomiálním čase
- nejznámější aproximace: *GT*
- přesný výpočet pomalý, pro velké *N* nemožný => v praxi se *MWT* neužívá

Příklad, kde *DT* <> *MWT*
<> *GT*





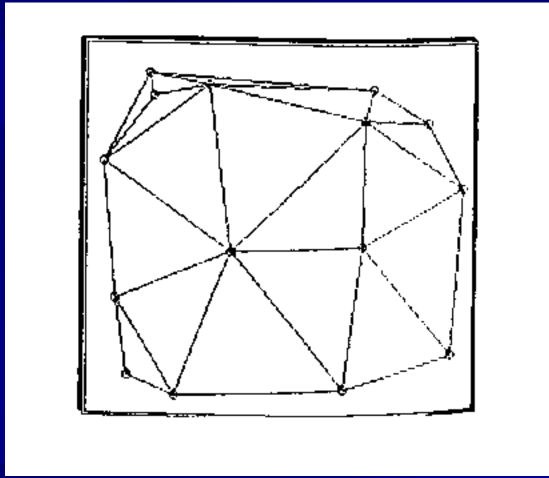




DT (plná čára) *MWT* *GT*

Metody konstrukce:

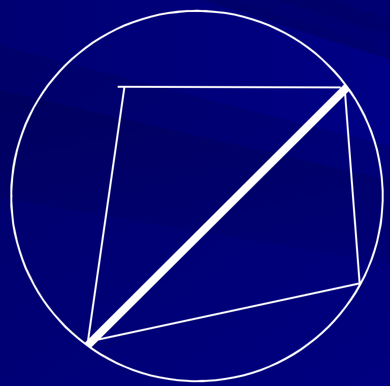
- brutální síla
- polynomiální řešení pro speciální datové množiny
- podgraf MWT
- přibližné řešení



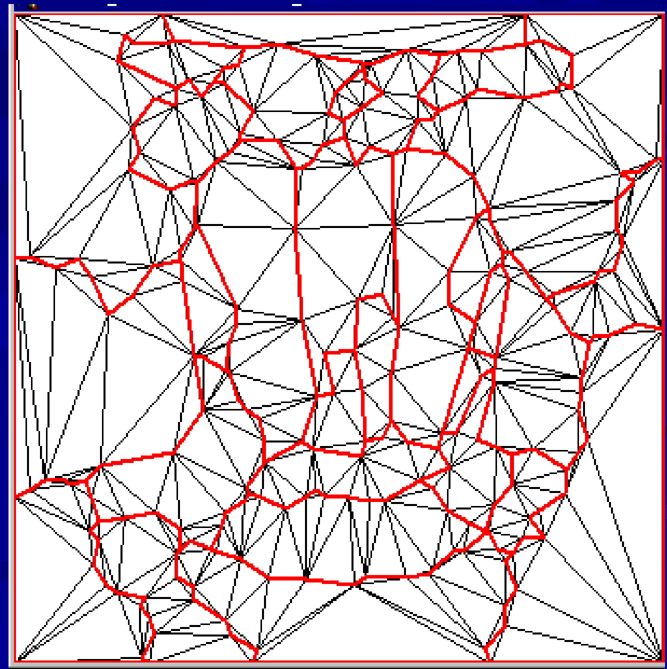
Příklad *MWT* pro *N*=15

6. Triangulace s omezením (*CDT*, *CGT*)

- do triangulace vnuceny povinné hrany
- *CGT*: napřed povinné hrany, pak greedy algoritmus
- *CDT*: nutná modifikace definice



přes povinnou hranu „nevidím“
=> neswapuji



7. Datově závislé triangulace (*DDT*)

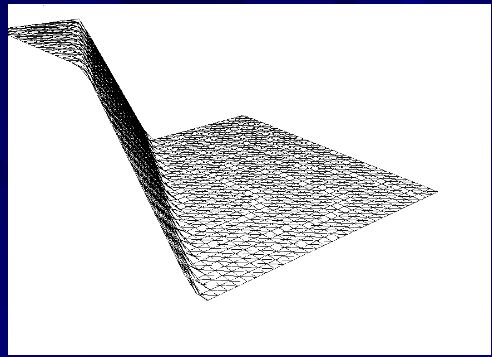
- bere v potaz také výšku bodů
- např. úhel mezi normálami 2 sousedních trojúhelníků
- minimalizovat sumu:

$$ABN = \arccos \frac{\boldsymbol{n_1 n_2}}{|\boldsymbol{n_1}||\boldsymbol{n_2}|}$$

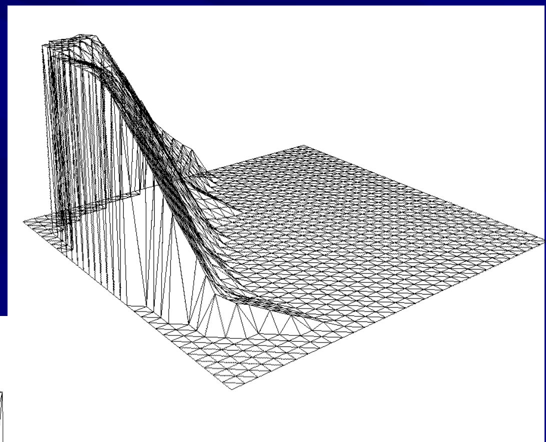
$$ABN_{sum} = \sum_{i=1}^{nE} (ABN_i)^2$$

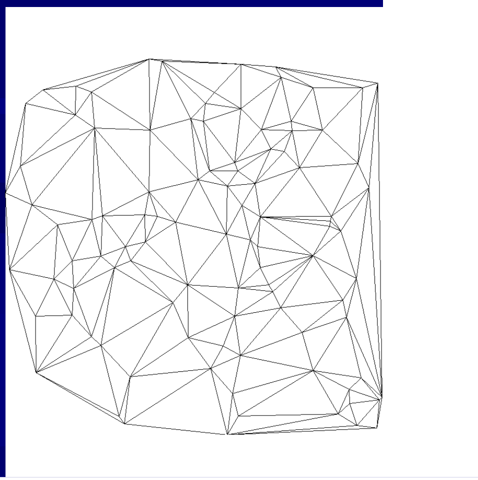
n_1, n_2 – normály soused. trojúhelníků,
nE – počet vnitřních hran triangulace

DDT se hodí pro:

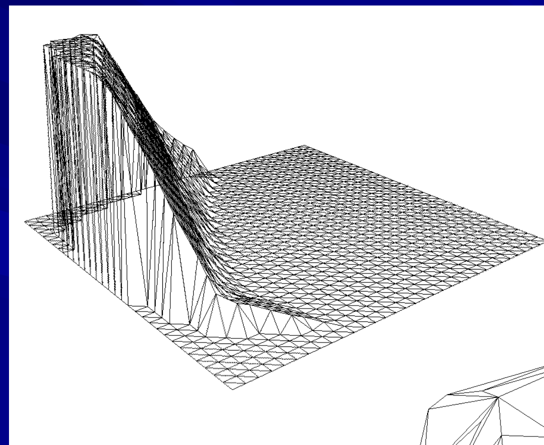


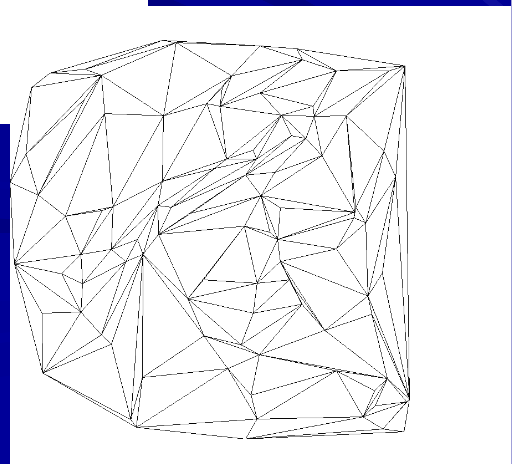
- triangulace bez uvažování výšky nerespektuje ostrý zlom





- s uvažováním výšky



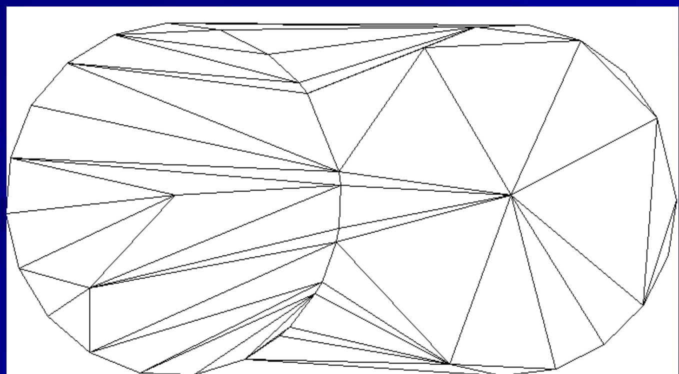


8. Multikriteriálně optimalizované triangulace

- kombinace různých kritérií lokálních i globálních
- genetická optimalizace – není zaručen úspěch

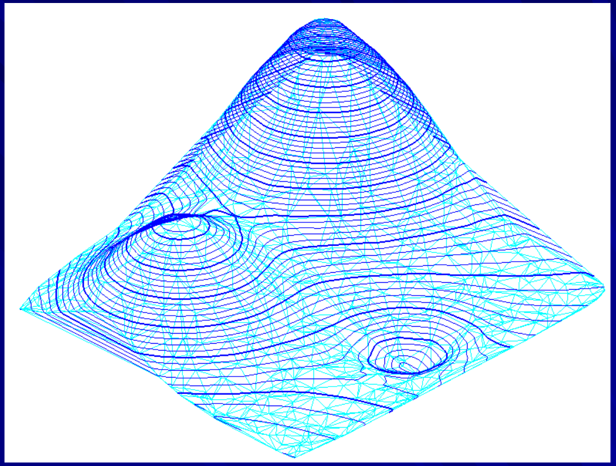
=> vzít 1 z počátečních triangulací DT nebo GT se známými vlastnostmi, genetika vylepší nebo ne ☺

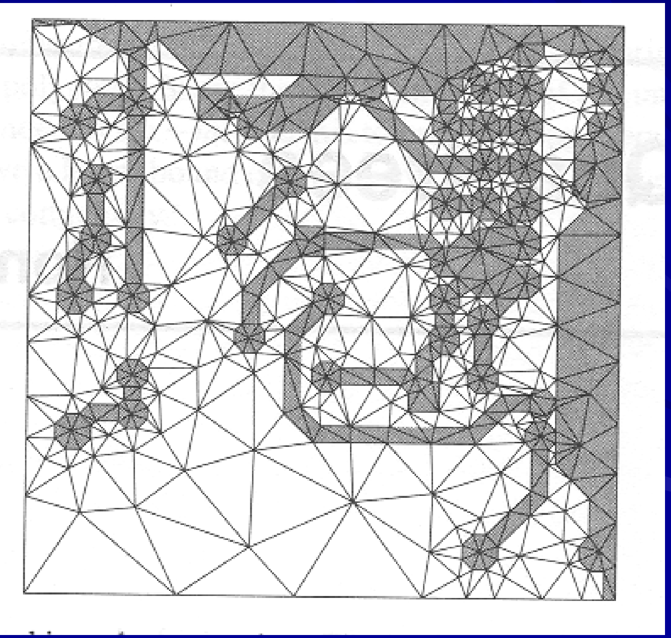
- velká volnost kritérií, ale velmi pomalé



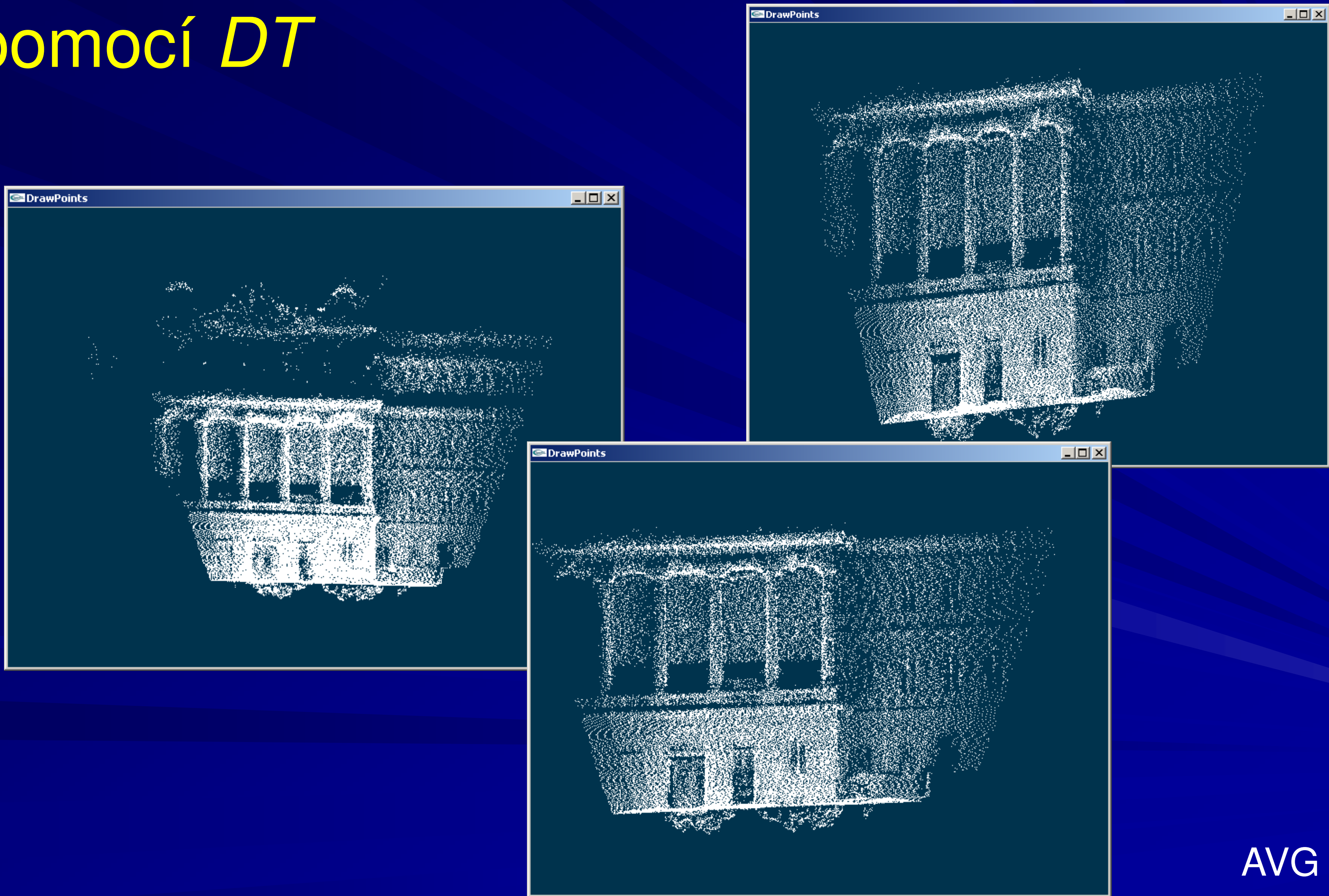
9. Aplikace

- GIS
- vizualizace a interpolace vědeckých dat
- robotika
- rozpoznávání obrazů
- trojúhelníkové sítě pro FEM
- počítačové sítě (min.kostra)
- přírodní vědy
- problém obrazové galerie …



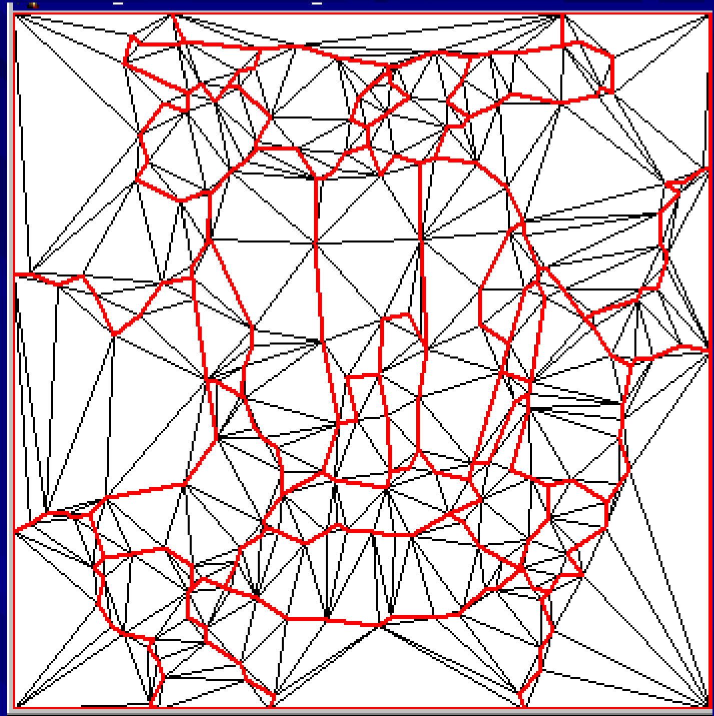


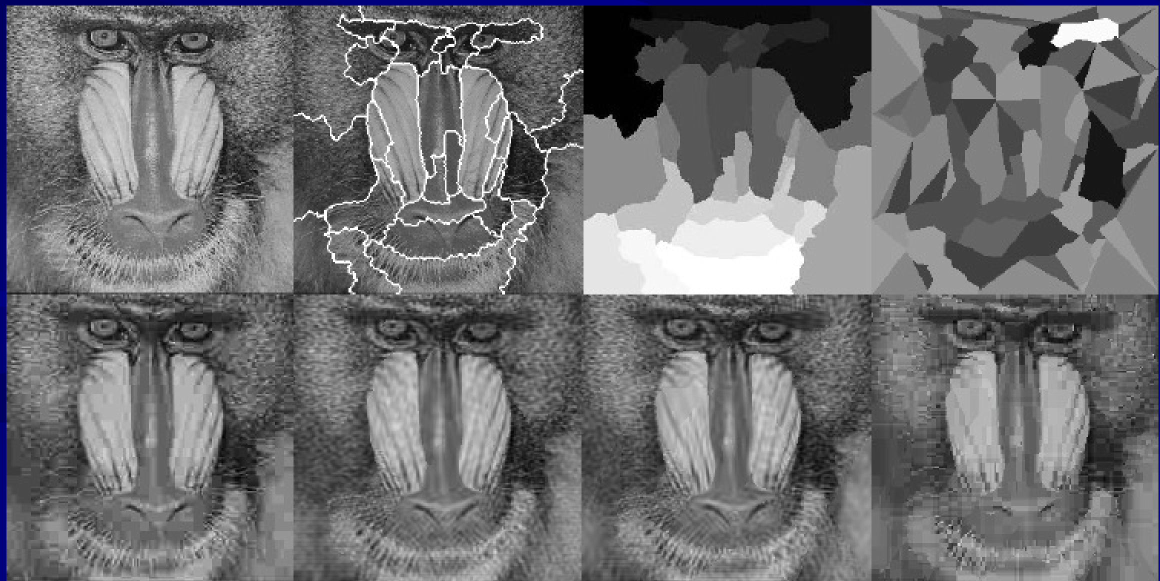
Odstranění „outliers“ bodů pomocí *DT*



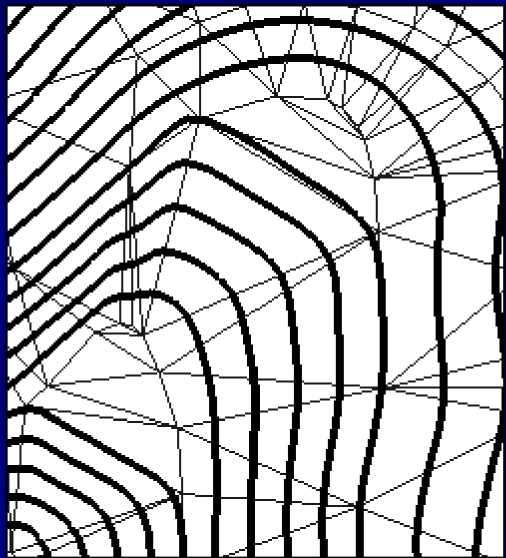
Triangulace oblastí v dig.obrazu pro aproximaci

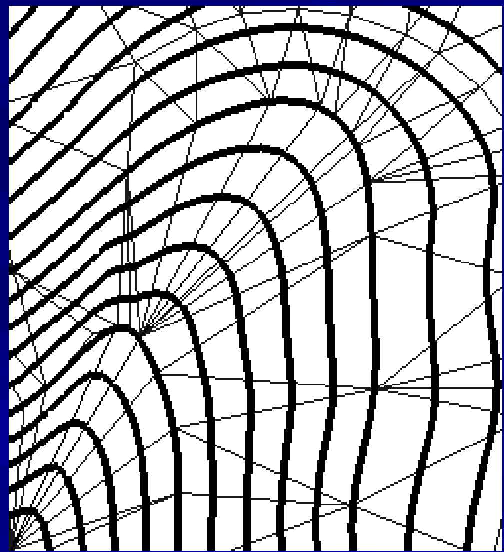
Více informací: J.Polec, M. Partyk





Vylepšování vrstevnic pomocí constraints





Vložení constrained edge zlepší tvar