Diskrétne Geometrické Štruktúry

8. Proximity Graphs

Martin Samuelčík

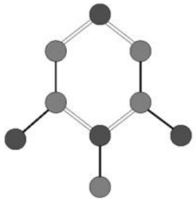
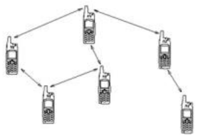
samuelcik@sccg.sk, www.sccg.sk/~samuelcik, I4

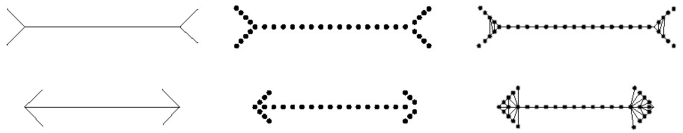
Mračná bodov

- Jednoduchá množina bodov
- Výstup z mnohých meracích zariadení – skenery, senzory, ...
- Rozvíjajúca sa oblasť v modelovaní a vizualizácii
- Potreba definovania povrchu, topológie

Proximity Graphs

- Grafy susednosti
- Prináša susednosť, topológiu pre neštruktúrované dáta
- Geometrické grafy – topológia je daná geometriou
- Použitie vo výpočtovej geometrii, počítačové videnie, tvorba sietí, biológia,....



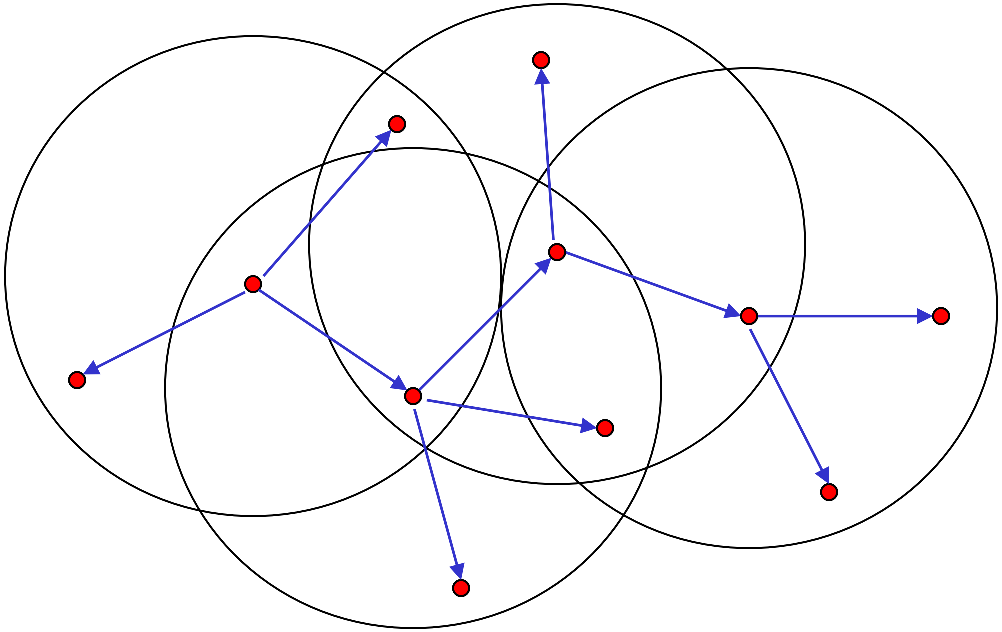
Definície

- Pracujeme v nejakom metrickom priestore
- Najčastejšie s Euklidovskou metrikou
- Najčastejší vstup – množina bodov $V \in R^d$
- Hrany spájajú vrcholy, ktoré sú si navzájom blízke ("proximity")
- Tým je definovaná topológia

$$d(\mathbf{x},\mathbf{y}) := \|\mathbf{x}-\mathbf{y}\|_p = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p}$$

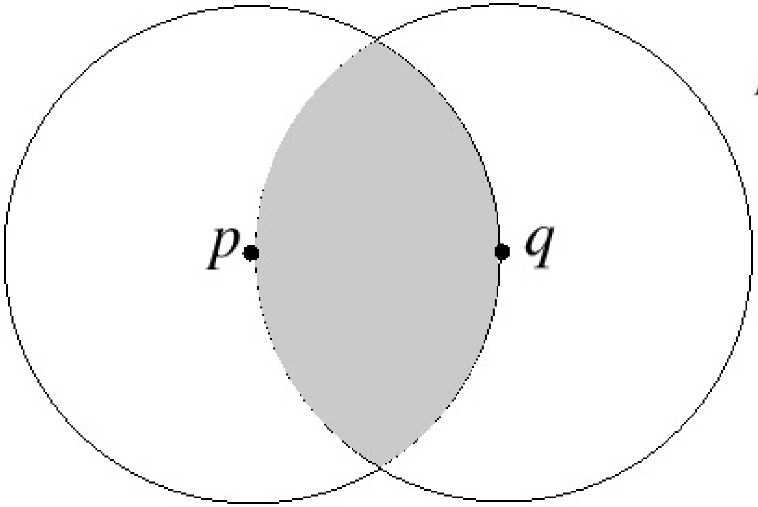
Jednoduchý graf

- Graf sfér s jednotkovou dĺžkou
- Unit Disc Graph, UDG(V)
- Množina hrán je $E := \{pq \mid d(p,q) \leq 1\}$;
- Vhodné pre siete mobilných vysielačov



Relatívny graf susednosti

- The relative neighborhood graph RNG(V)
- L(p, q) = B(p, d) ∩ B(q, d), kde d = ||p – q||
- Množina hrán $E := \{pq \parallel L(p,q) \cap V = \emptyset\}$.



$$pq \in E \Leftrightarrow \nexists v \in V : d(p,v) < d(p,q) \ \wedge \ d(q,v) < d(p,q).$$

$$pq \in E \Leftrightarrow \forall v \in V : d(p,q) \leq \max\{d(p,v), d(q,v)\} .$$

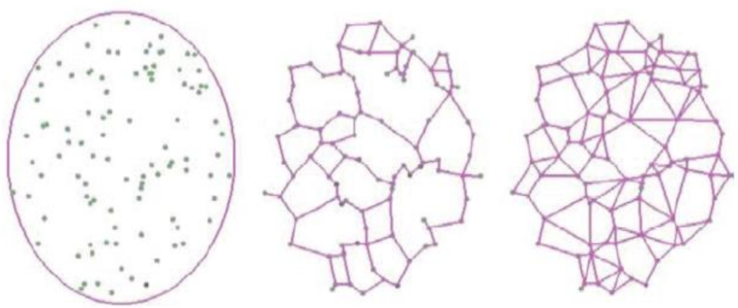
Gabrielov graf

- Definované pomocou sféry, spojitý graf
- $G(p,q) := B\left(\frac{\mathbf{p}+\mathbf{q}}{2}, \frac{d}{2}\right)$ kde d = ||p − q||
- GG(V) má množinu hrán

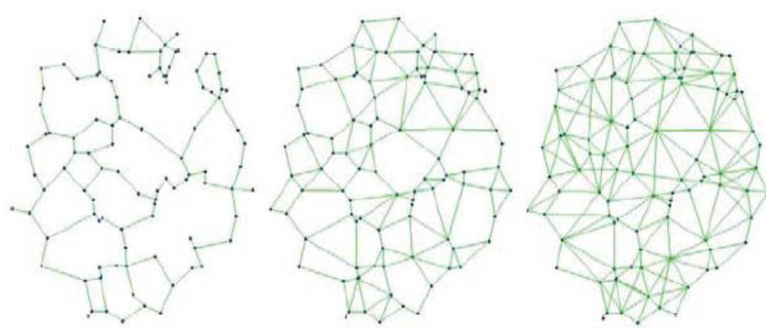
$$E := \{pq \parallel G(p,q) \cap V = \emptyset\}.$$

$$pq \in E \Leftrightarrow \forall v \in V : d(p,q) \leq \sqrt{d^2(p,v) + d^2(q,v)}.$$

- Rozšírenie v podobe elíps (EGG)



(a) Input points (b) RNG (c) GG



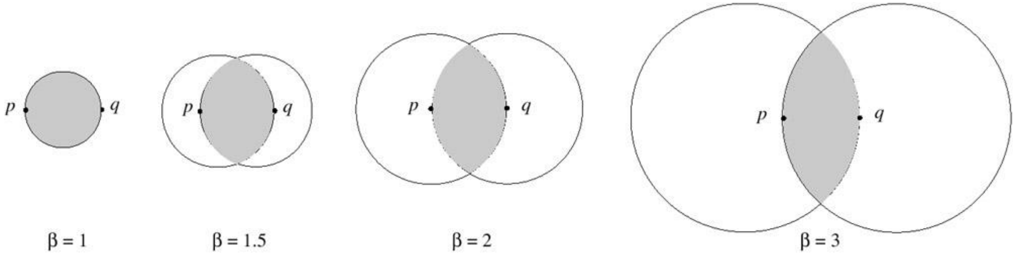
(a) EGG(2.0) (b) EGG(1.0) (c) EGG(0.7)

β-kostry

- Grafy dané parametrom β, $1 \leq \beta < \infty$.
- Zovšeobecnenie, označenie $BG_\beta(V)$
- Okolie bodov p a q je dané

$$U_\beta(p,q) := B\left((1-\frac{\beta}{2})\mathbf{p}+\frac{\beta}{2}\mathbf{q}, \frac{\beta}{2}d\right) \cap B\left((1-\frac{\beta}{2})\mathbf{q}+\frac{\beta}{2}\mathbf{p}, \frac{\beta}{2}d\right),$$

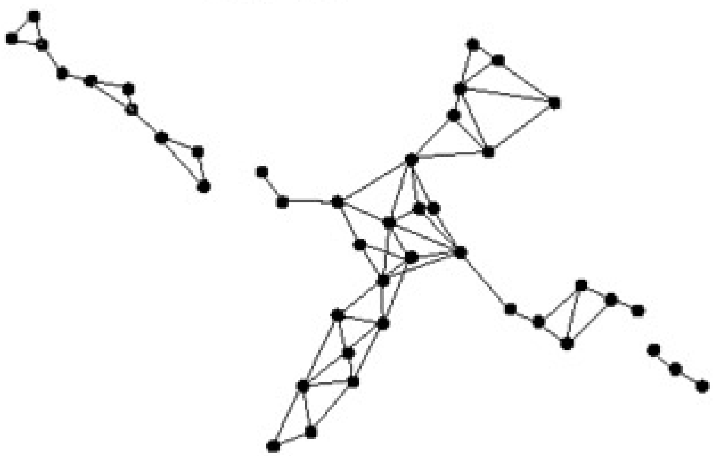
- Množina hrán $E := \{pq \parallel U_\beta(p,q) \cap V = \varnothing\}$.
- $RNG(V) = BG_2(V)$, $GG(V) = BG_1(V)$
- Ak $\beta_1 > \beta_2$, potom $BG_{\beta_1}(V) \subset BG_{\beta_2}(V)$

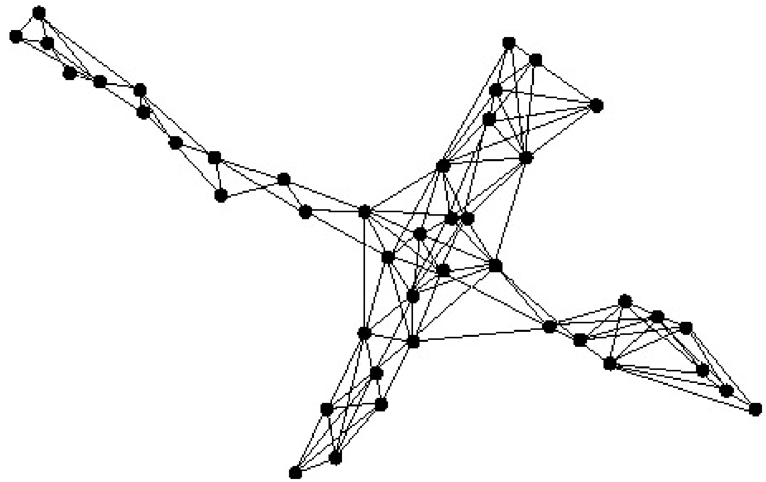


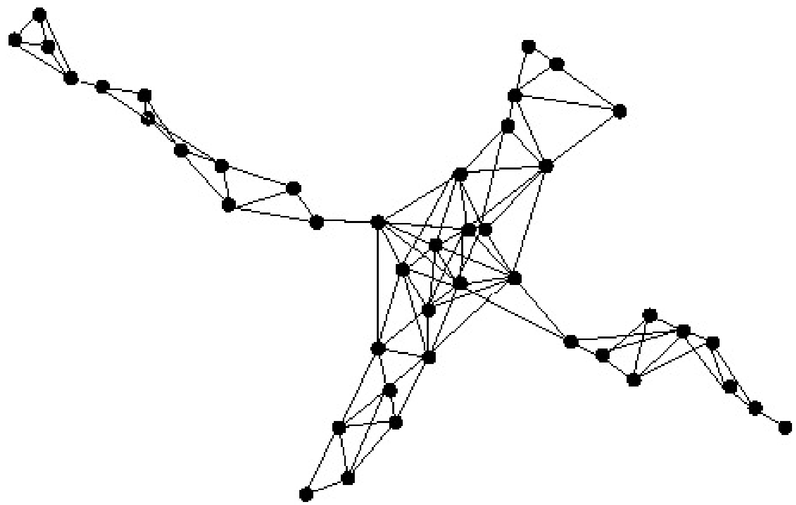
Sféry vplyvu

- Sphere-of-influence graph, SIG(V)
- Pre bod $p \in V$, r_p je vzdialenosť k najbližšiemu susedovi
- Množina hrán $E := \{pq \parallel d(p,q) \leq r_p + r_q\}$.
- Spája body, ktoré sú „blízko" vzhľadom na lokálnu hustotu bodov
- Rozšírenie, r-SIG, hľadá najkratšiu cestu z bodu p, ktorá ma r-hrán, r_p je vzdialenosť k tomu bodu

SIG, 3-SIG, 3-SIG bez dlhých hrán

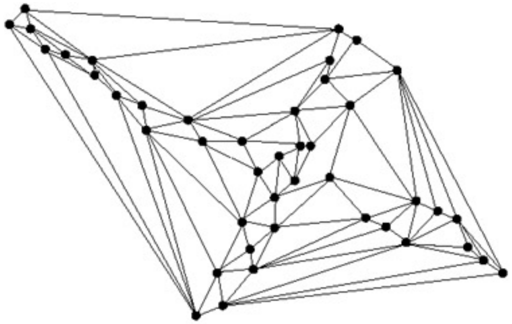




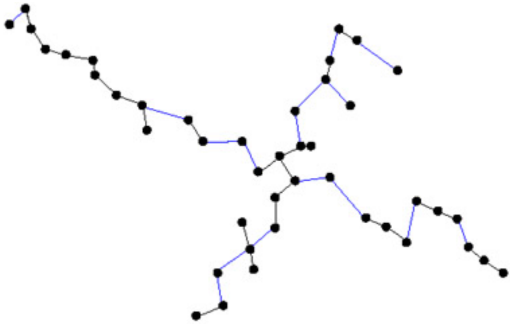


Geometrické grafy

- *Minimum Spanning Tree* (MST) spája body do stromu tak, aby súčet dĺžok hrán bol minimálny
- *Delaunay graph* (DG) spája dva body p a q vtedy, ak existuje kružnica k, ktorá prechádza bodmi p, q a žiadne iné body nie sú vnútri kružnice k.



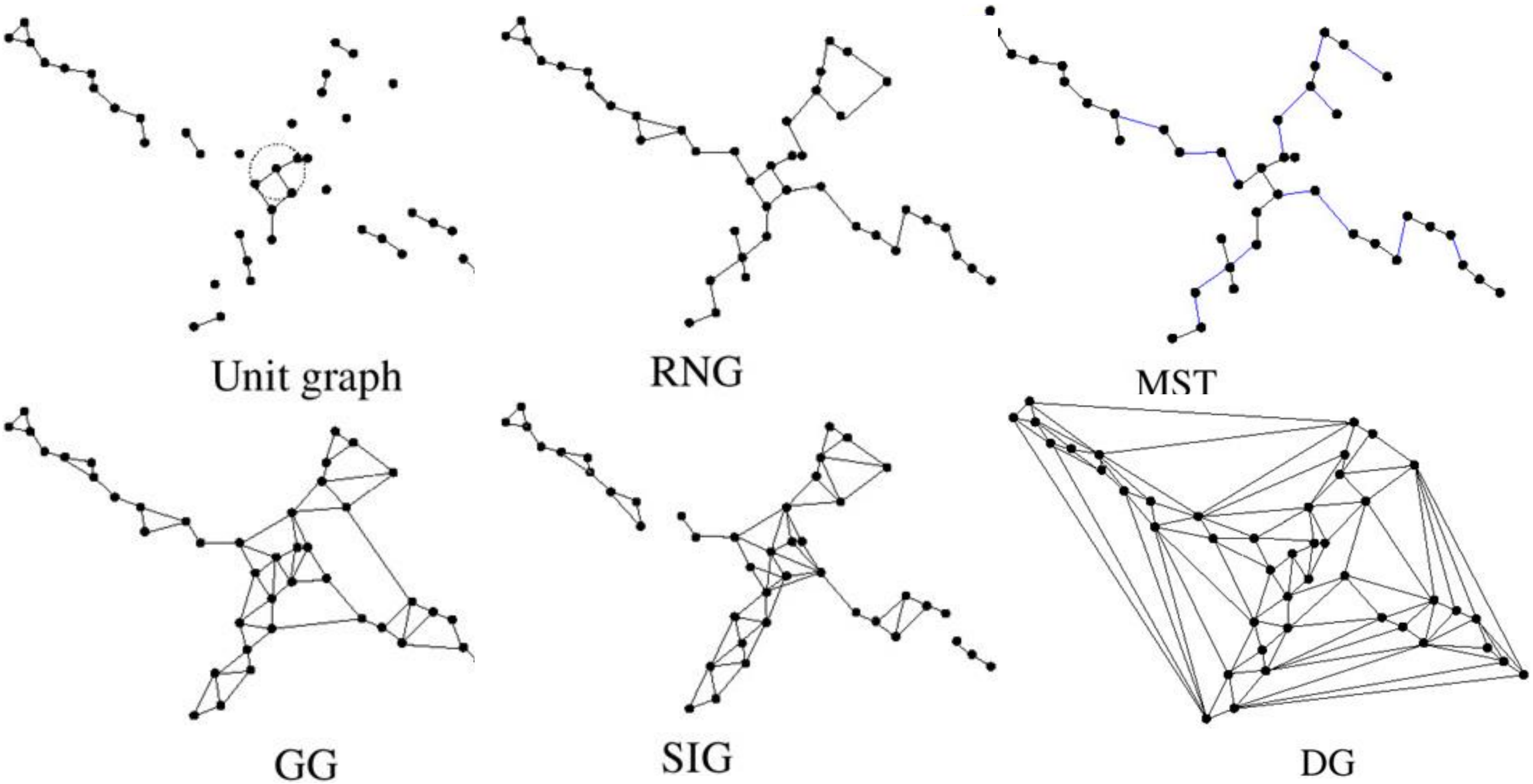
DG



MST

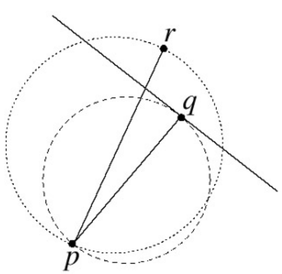
Porovnanie

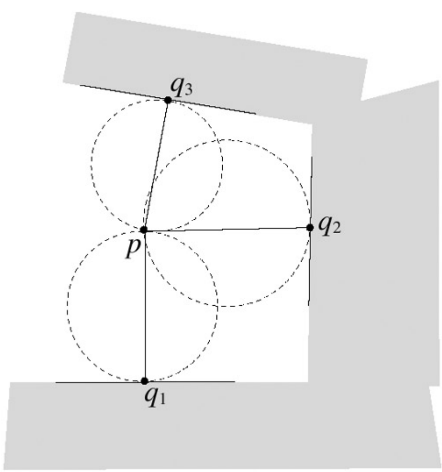
- |MST(V)| ≤ |RNG(V)| ≤ |GG(V)| ≤ |DG(V)|
- $MST(V) \subseteq RNG(V) \subseteq GG(V) \subseteq DG(V).$



Konštrukcia GG

- Brute-force algoritmus pre GG -> $O(n^3)$
- Zlepšenie-heuristika-odstránenie vrcholov napravo od skúmaného vrcholu - $O(n^2)$
- Z DG odstraňovaním hrán - $O(n^2)$






```
CreateGG(V)
{
        for (all p from V)
        {
                N_p = V \ p;
                for (all q from N_p)
                {
                        for (all r in V)
                        {
                                if (d(p,r)^2+d(q,r)^2 <= d(p,q)^2)
                                                remove q from N_p;
                                else if (r is to the right of H_qp)
                                                remove r from N_p;
                        }
                }
                Connect p with points from N_p;
        }
}
```

Konštrukcia r-SIG

- V priemere lineárna časová zložitosť
- Pamäťová zložitosť je tiež lineárna

```
CreaterSIG(r, V)
{
        initialize grid with n cells;
        for (all p in V)
             assign p to its grid cell;
        for (all p in V)
             find r-th nearest neighbor to p by searching the grid cells in spiral order around p with increasing
             distance;
        for (all p in V)
        {
                for (all cells around p that intersect the sphere of influence around p (in spiral order))
                      assign p to cell;
        }
        for (all cells in the grid)
        {
                for (all pairs p_i, p_j of points assigned to the current cell)
                {
                        if (spheres of influence of p_i and p_j intersect)
                              create edge p_ip_j;
                }
        }
}
```

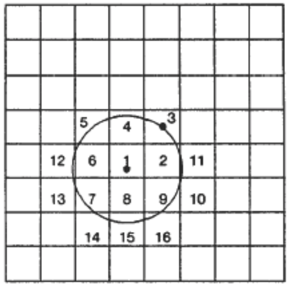


Fig. 1 Spiral nearest neighbor search using cells.

Voronoiov diagram

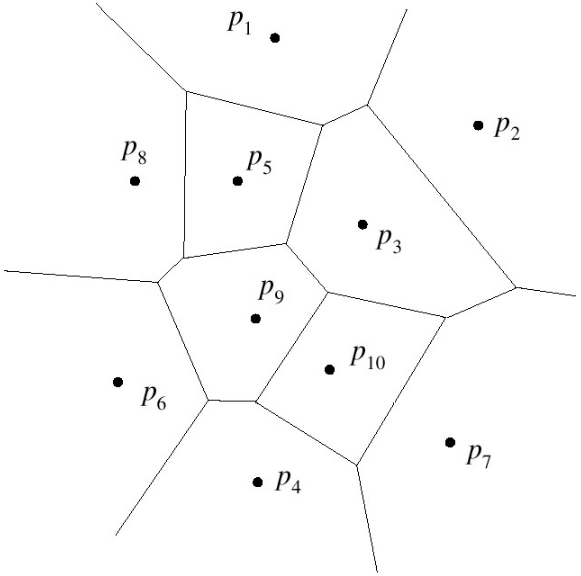
- Geometrický graf
- Pre danú množinu bodov S={p_1, p_2, ..., p_n}, každá stena VD je priradená jednému bodu p_i a pre každý bod oblasti je tento bod bližšie k p_i ako ku inému bodu z S

$$\mathrm{Bis}(p_i, p_j) = \{x \mid d(p_i, x) = d(p_j, x)\}$$

$$\mathrm{H}(p_i, p_j) = \{x \mid d(p_i, x) < d(p_j, x)\}$$

$$\mathrm{VoR}(p_i, S) = \bigcap_{p_j \in S, p_j \neq p_i} H(p_i, p_j)\,.$$

$$\mathrm{VD}(S) := \bigcup_{p_i, p_j \in S, p_i \neq p_j} \overline{\mathrm{VoR}(p_i, S)} \cap \overline{\mathrm{VoR}(p_j, S)}.$$

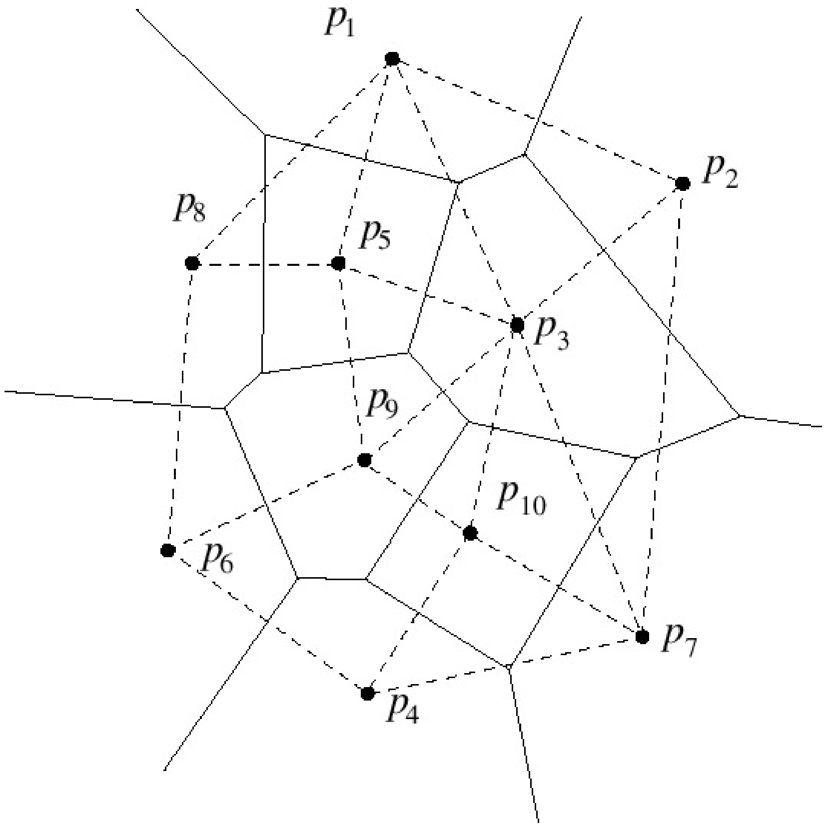


Vlastnosti VD

- VoR(p_i, S) je prienikom max n-1 polrovín, je to otvorená a konvexná množina
- VoR(p_i, S) je ohraničný práve vtedy keď p_i patrí do vnútra konvexného obalu S
- VD má O(n) hrán a vrcholov
- Priemerný počet hrán na hranici VoR(p_i, S) je 6

Delaunayova triangulácia

- Pre množinu bodov S a VD(S), DT(S) je duálny graf k VD
- Body p_i, p_j, p_k tvoria trojuholník v DT $\Leftrightarrow$ kružnica prechádzajúca tromi bodmi neobsahuje vnútri ani na hranici iný bod z S
- DT maximalizuje minimálny vnútorný uhol

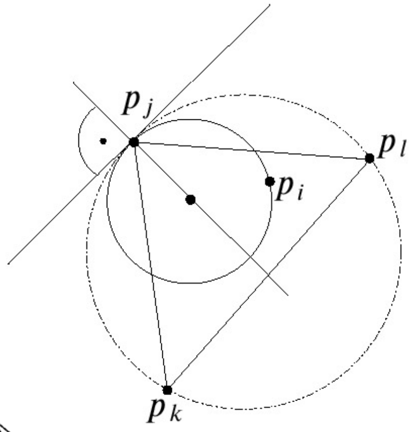


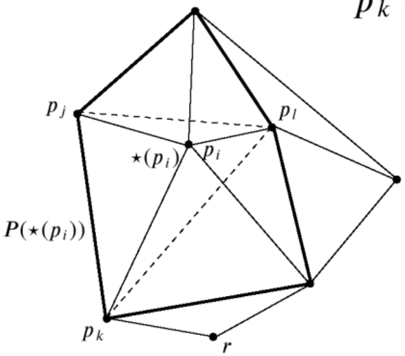
Konštrukcia DT

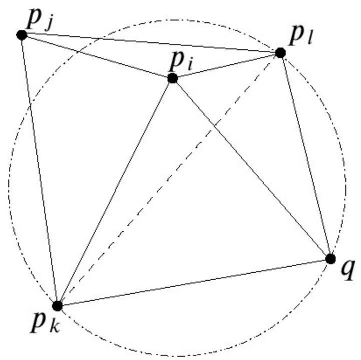
- Algoritmus vkladania nového bodu p do DT
- Dva prípady – bod vložíme do konvexného obalu S alebo mimo
- Po vložení sa kontroluje, či sa neporušila vlastnosť DT
- Pri porušení as vykonáva otočenie hrany v „zlom“ štvoruholníku
- Zložitosť $O(n^2)$ v najhoršom prípade, $O(n\log(n))$ v priemernom prípade, može byť rozšírené na $O(n\log(n))$ pre všeobecý prípad

1. prípad

- p_i leží v $T = \Delta(p_j, p_k, p_l)$
- Hrany p_ip_j, p_ip_k, p_ip_l patria novej DT
- Konflikt môže nastať v Δ ktoré sú susedné s T
- Vtedy sa otočí hrana T
- Konflikt aj v ďalších Δ
- Ak hrany $P(*(p_i))$ sú z DT, koniec

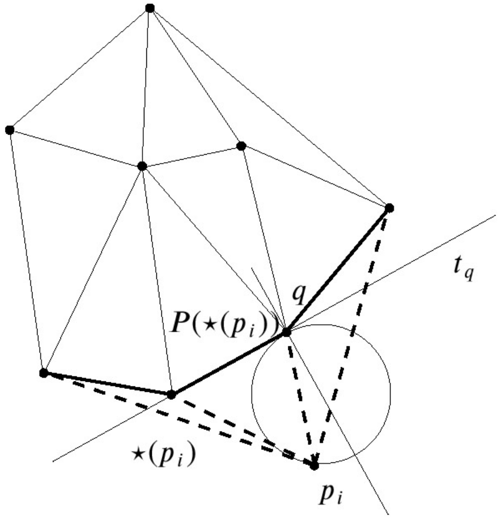






2. prípad

- Ak *p* nepatrí do konvexného obalu pôvodnej DT
- Pre všetky body *q* z S, ktoré sú „viditeľné" z *p*, *pq* bude patriť novej DT
- Niektoré hrany na hranici bude treba otočiť



Algoritmus

```
Delaunay(S)  (S is set of sites)
{
        T = new array;
        while (S.size() > 0)
        {
                p = S.First;
                S.DeleteFirst;
                T.InsertSite(p);
        }
}
```

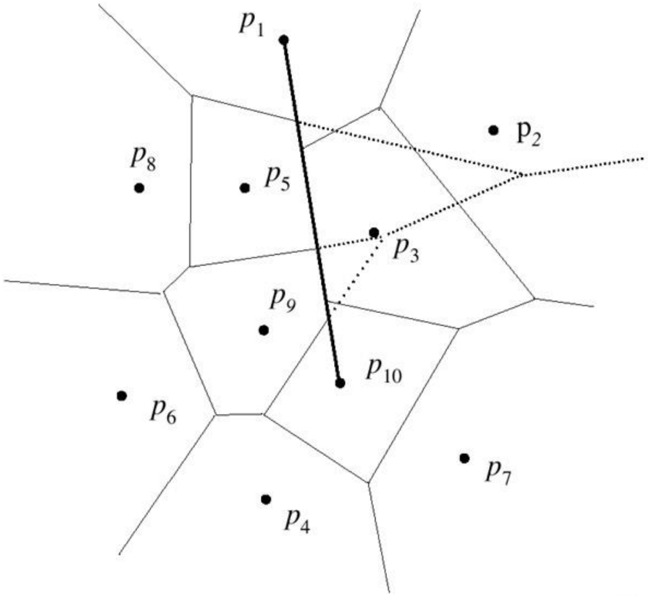
```
InsertSite(T, p) (T represents the current Delaunay triangulation, p is a new site)
{
        t = T.FindTriangle(p);
        if (t != NULL)
              Star(p) = t.CreateStar(p);
        else
              Star(p) := T.HullEdges(p);
        T.Insert(Star(p), t);
        StarPoly = t.Edges();
        while (StarPoly.size() > 0)
        {
                e = StarPoly.First();
                StarPoly.DeleteFirst();
                q = p.Opposite(e);
                if (q ≠ NULL)
                {
                        (r, s) = e.EndPoints();
                        if (InCircleTest(p, r, s, q))
                        {
                                T.Remove(e);
                                T.Add((p, q));
                                StarPoly.Add((r,q));
                                StarPoly.Add((s,q));
                        }
                }
        }
}
```

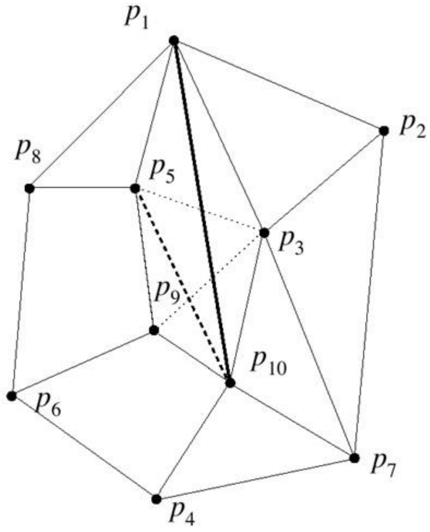
Zovšeobecnenia VD, DT

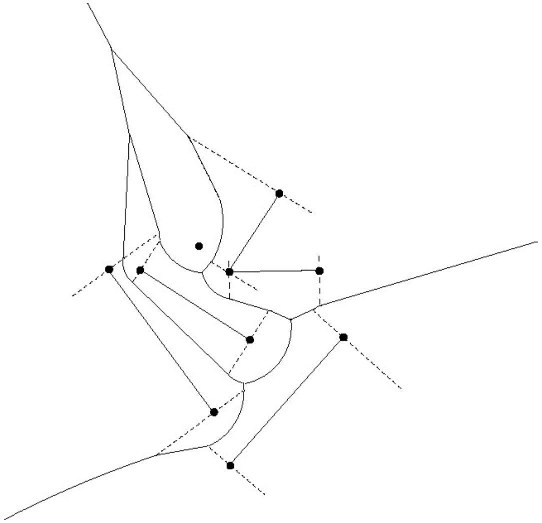
- Rozšírenie do vyšších dimenzií, 3D – tetrahedralizácia
- Použitie obmedzení – sú dané časti (hrany, …), ktoré triangulácia obsahuje (Constrained DT), obmedzujúca metrika pre VD
- Použitie iných metrík
- Váhovanie $w(p_i)d(p_i, q) = w(p_j)d(p_j, q)$
- Rozšírenie bodov na zložitejšie objekty

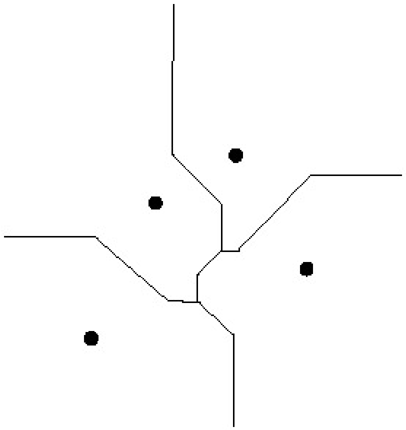
$$b(x, y) = \begin{cases} d(x, y), & \text{if } \overline{xy} \cap L = \emptyset, \\ \infty, & \text{otherwise}, \end{cases}$$

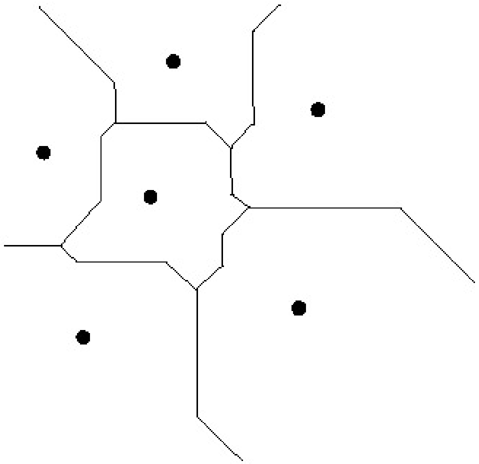
Zovšeobecnenia

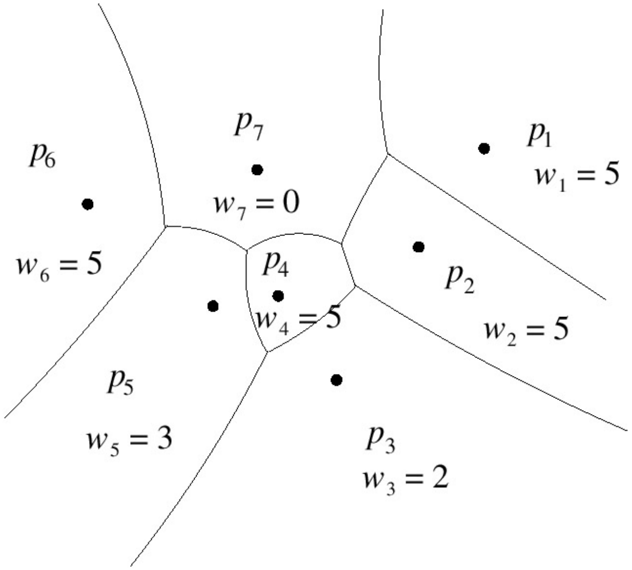












Definovanie povrchu

- Z množiny bodov implicitný povrch, ktorý minimalizuje akúsi vzdialenostnú funkciu
- Weighted Least Squares (WLS)
- Povrch S je určený funkciou f(x)
- $S = \{x | f(x) = 0\}$

$$f(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot (\mathbf{a}(\mathbf{x}) - \mathbf{x}),$$

$$\mathbf{a}(\mathbf{x}) = \frac{\sum_{i=1}^{N} \theta(\|\mathbf{x} - \mathbf{p}_i\|)\mathbf{p}_i}{\sum_{i=1}^{N} \theta(\|\mathbf{x} - \mathbf{p}_i\|)}.$$

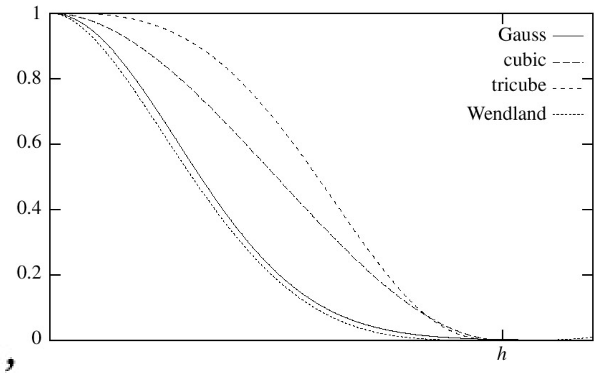
Váhovacie funkcie

- Určujú váhu bodu podľa vzdialenosti
- Gaussian

$$\theta(d) = e^{-d^2/h^2}, \quad d = \|\mathbf{x} - \mathbf{p}\|,$$

- Cubic

$$\theta(d) = 2\left(\frac{d}{h}\right)^3 - 3\left(\frac{d}{h}\right)^2 + 1,$$



- Tricubic

$$\theta(d) = 2\left(\frac{d}{h}\right)^3 - 3\left(\frac{d}{h}\right)^2 + 1,$$

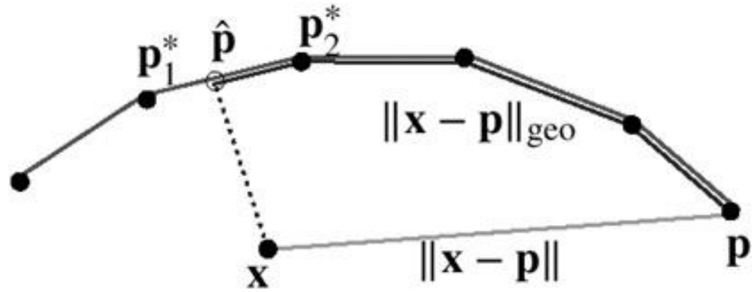
- Wendland

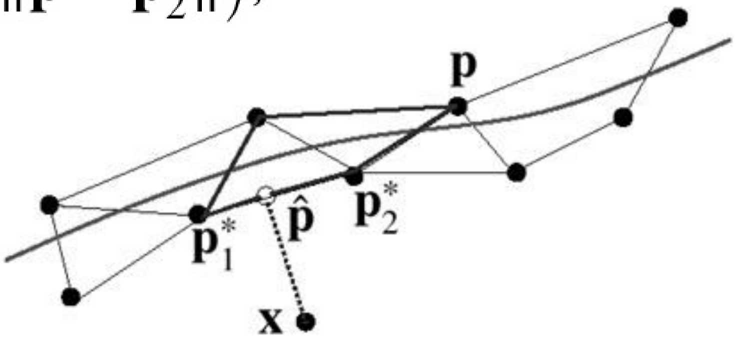
$$\theta(d) = \left(1 - \frac{d}{h}\right)^4 \left(4\frac{d}{h} + 1\right),$$

Využitie grafov

- Problémy v zle vzorkovaných oblastiach s Euklid. vzd.
- Geodedická vzdialenosť na základe aproximácie povrchu grafom geometrickej susednosti (SIG, …)

$$d_{\mathrm{geo}}(\mathbf{x}, \mathbf{p}) = (1-a)\Big(d(\mathbf{p}_1^*, p) + \|\hat{\mathbf{p}} - \mathbf{p}_1^*\|\Big) + a\Big(d(\mathbf{p}_2^*, p) + \|\hat{\mathbf{p}} - \mathbf{p}_2^*\|\Big),$$





Parametre

- Normála
  - Minimalizácia $$\sum_{i=1}^{N}\left(\mathbf{n}(\mathbf{x})\cdot(\mathbf{a}(\mathbf{x})-\mathbf{p}_i)\right)^2\theta(\|\mathbf{x}-\mathbf{p}_i\|) \qquad \|\mathbf{n}(\mathbf{x})\|=1.$$
  - Najmenší vlastný vektor matice

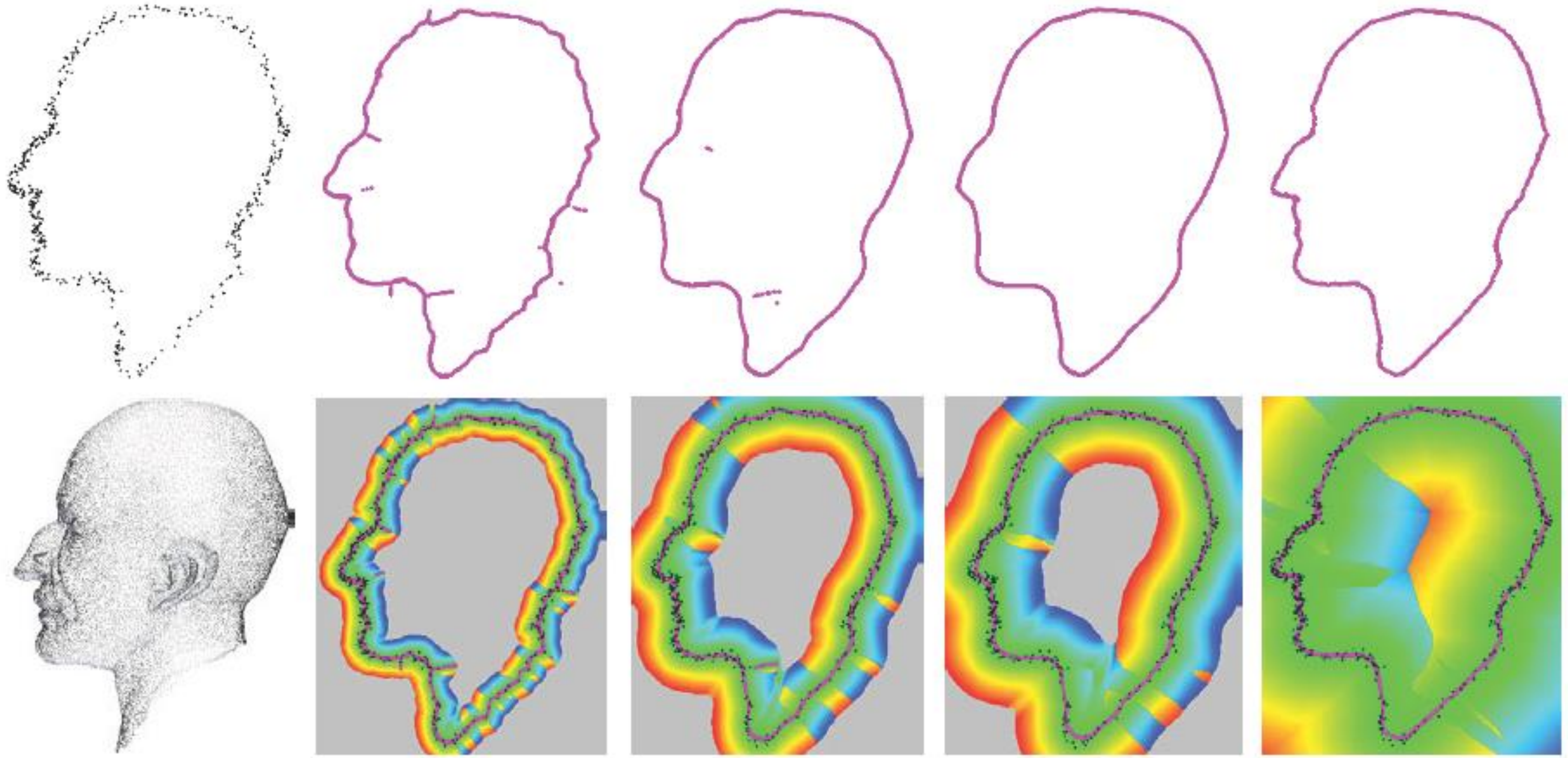
$$b_{ij}=\sum_{k=1}^{N}\theta(\|\mathbf{x}-\mathbf{p}_k\|)(p_{k_i}-a(\mathbf{x})_i)(p_{k_j}-a(\mathbf{x})_j).$$

- Vplyv vzdialenosti *h*
  - Vhodné použitie adpatívneho *h*
  - Použitie geodetickej vzdialenosti

$$r(\mathbf{x})=\frac{1}{r}\cdot\frac{\|\hat{\mathbf{p}}-\mathbf{p}_2^*\|\cdot r_1+\|\hat{\mathbf{p}}-\mathbf{p}_1^*\|\cdot r_2}{\|\mathbf{p}_2^*-\mathbf{p}_1^*\|},$$

$$h(\mathbf{x})=\frac{\eta\, r(\mathbf{x})}{\sqrt{-\log\theta_\varepsilon}},$$

Výsledky



WLS, h=5 | WLS, h=10 | WLS, h=14 | SIG, autom. h

Výsledky

Otázky?