

Diskrétne geometrické štruktúry

9.

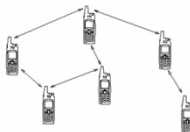
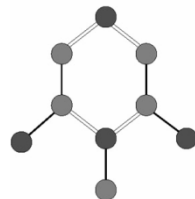
Martin Florek

florek@sccg.sk

www.sccg.sk/~florek

Proximity Graphs

- neighborhood graphs
- prináša susednosť, štruktúru, tvar, topológiu pre neštruktúrované dáta
- geometrické grafy – topológia je daná geometriou
- použitie vo výpočtovej geometrii, počítačové videnie, tvorba sietí, geografia, biológia,...

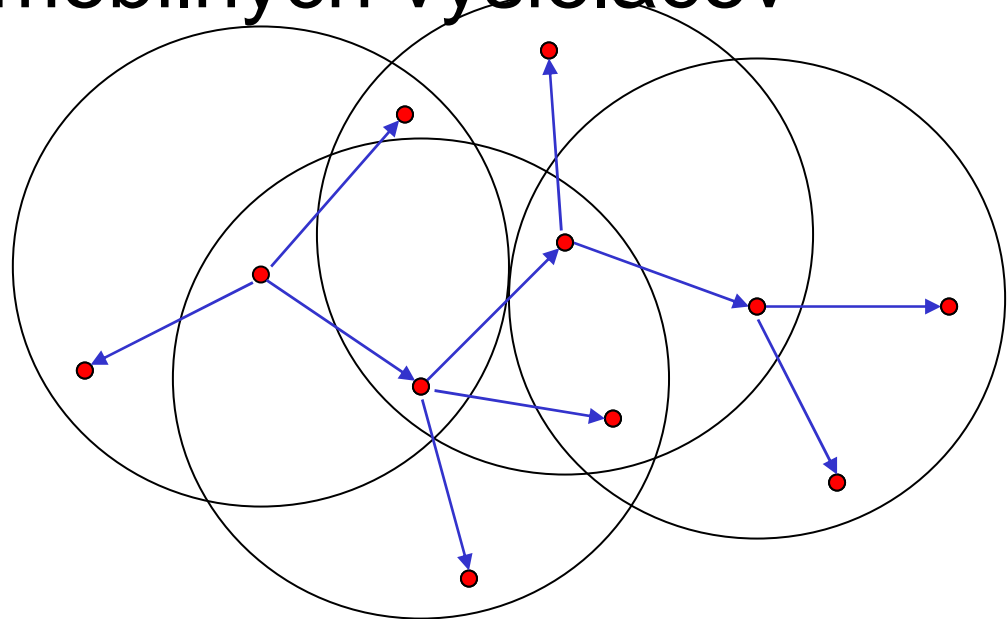


Definície

- 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

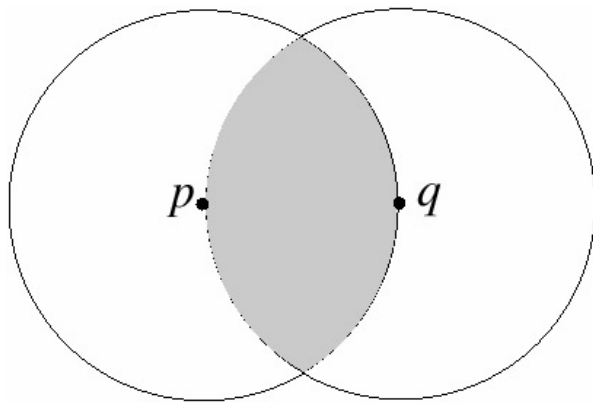
Jednoduché grafy

- Unit Disc Graph, UDG(V)
- graf sfér s jednotkovou délkou
- množina hrán je $E := \{pq \mid d(p, q) \leq 1\}$;
- vhodné pre siete mobilných vysieláčov



Relatívny graf susednosti

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



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

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

Gabrielov graf

- definované pomocou sféry

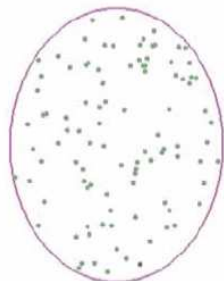
$$G(p, q) := B\left(\frac{p+q}{2}, \frac{d}{2}\right) \text{ kde } d = \|p - q\|$$

- GG(V) má množinu hrán

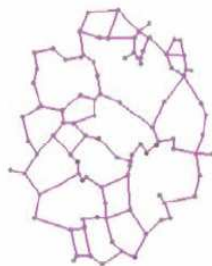
$$E := \{pq \mid G(p, q) \cap V = \emptyset\}.$$

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

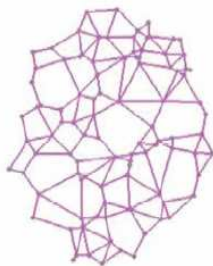
- rozšírenie v podobe elíps (EGG)



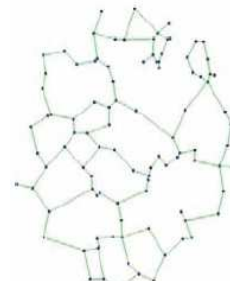
(a) Input points



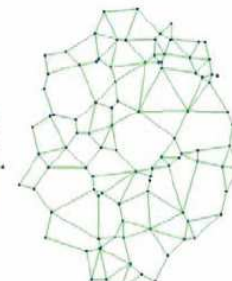
(b) RNG



(c) GG



(a) EGG(2.0)



(b) EGG(1.0)



(c) EGG(0.7)

β -kostry

- grafy dané parametrom β , $1 \leq \beta < \infty$

- označenie $BG_\beta(V)$

- okolie bodov p a q je dané

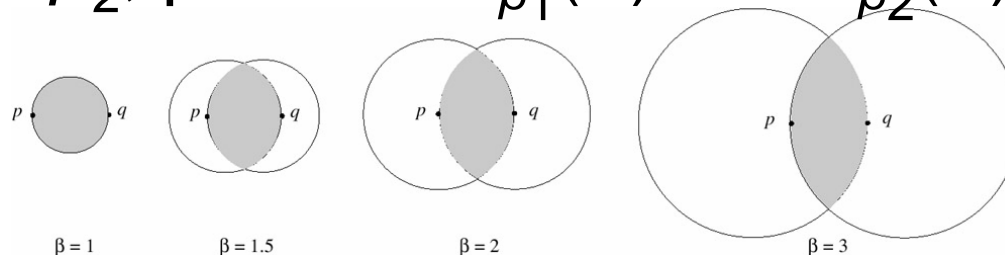
$$U_\beta(p, q) := B\left((1 - \frac{\beta}{2})\mathbf{p} + \frac{\beta}{2}\mathbf{q}, \frac{\beta}{2}d\right) \cap B\left((1 - \frac{\beta}{2})\mathbf{q} + \frac{\beta}{2}\mathbf{p}, \frac{\beta}{2}d\right),$$

- kde $d = \beta/2 \cdot \|p - q\|$

- množina hrán $E := \{pq \mid U_\beta(p, q) \cap V = \emptyset\}$.

- $RNG(V) = BG_2(V)$, $GG(V) = BG_1(V)$

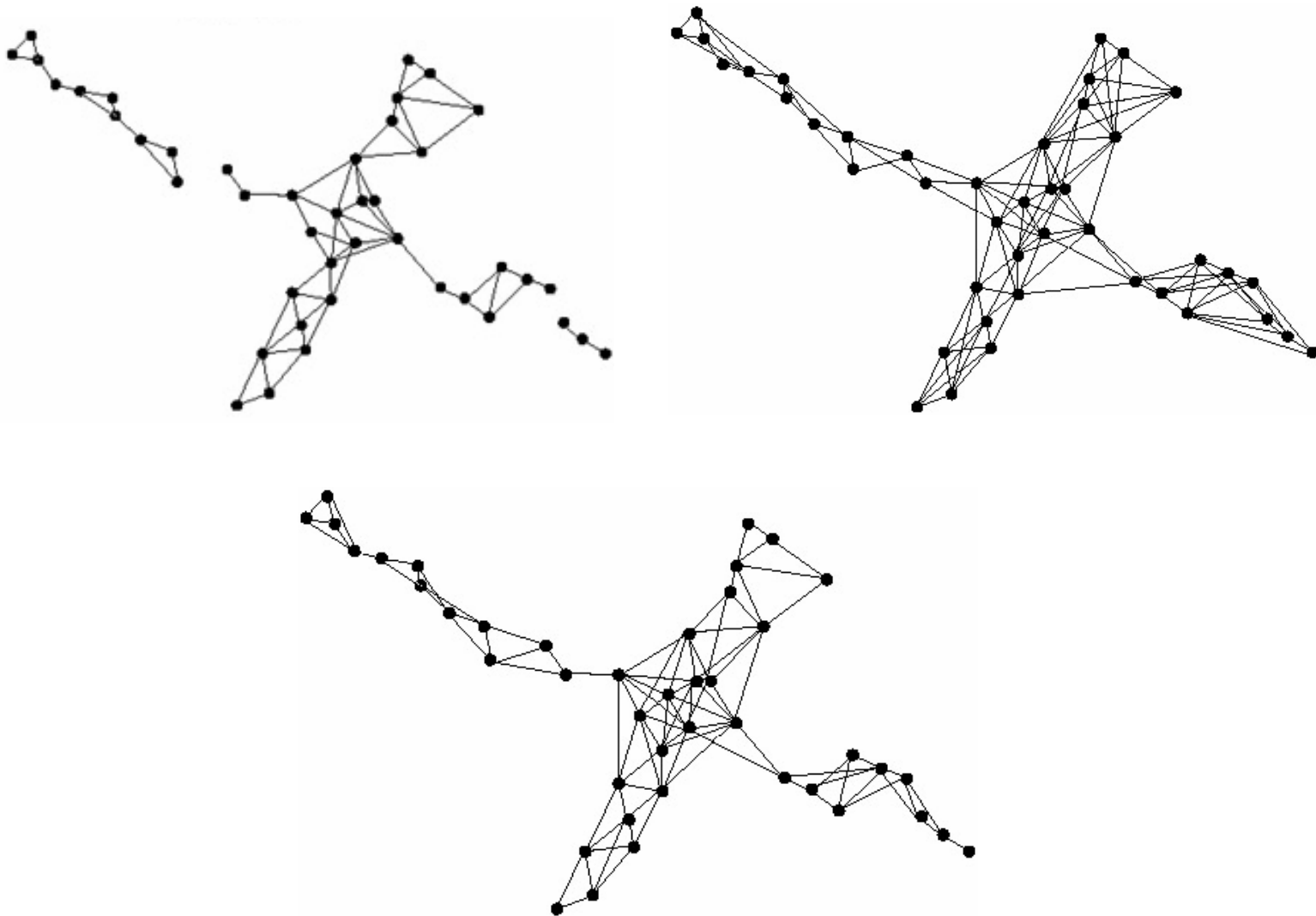
- ak $\beta_1 > \beta_2$, potom $BG_{\beta_1}(V) \subset BG_{\beta_2}(V)$



Sféry vplyvu

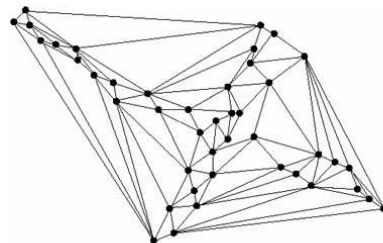
- sphere-of-influence graph, SIG(V)
- pre bod $p \in V$, r_p je vzdialenosť k najbližšiemu susedovi
- množina hrán $E := \{pq \mid d(p, q) \leq r_p + r_q\}$.
- nemusí byť spojitý →
 - rozšírenie: r -SIG, r_p je vzdialenosť k r -tému najbližšiemu susedovi

SIG, 3-SIG, 3-SIG bez dlhých hrán

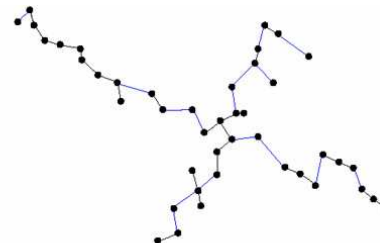


Geometrické grafy

- *Minimum Spanning Tree* (MST) spája body do stromu tak, aby súčet dĺžok hrán bol minimálny (minimálna kostra)
- *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



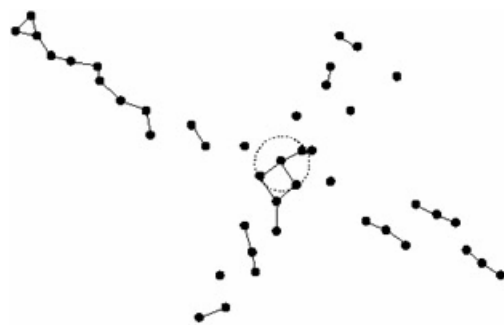
DG



MST

Porovnanie

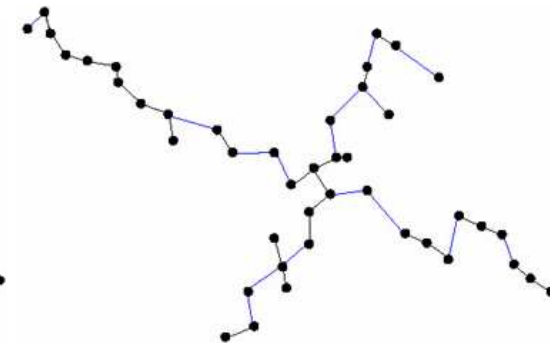
- $MST(V) \subseteq RNG(V) \subseteq GG(V) \subseteq DG(V)$.
- $|MST(V)| \leq |RNG(V)| \leq |GG(V)| \leq |DG(V)|$



Unit graph



RNG



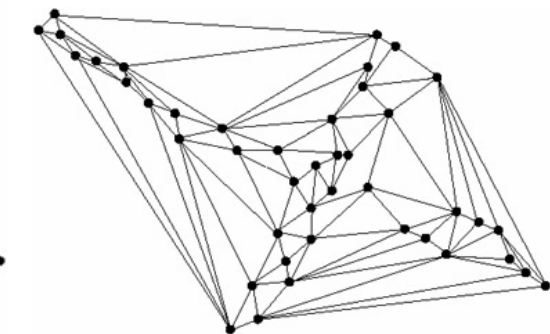
MST



GG



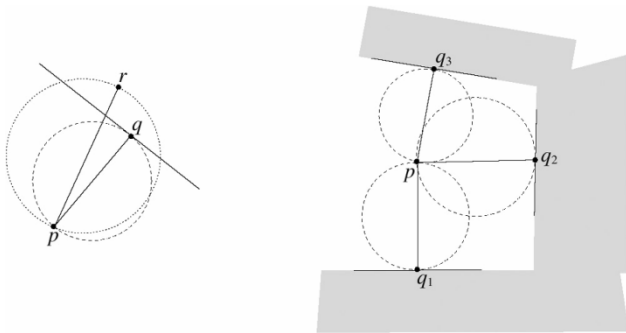
SIG



DG

Konštrukcia GG

- hrubá sila pre vytvorenie GG $\rightarrow O(n^3)$
 - všetky možné hrany $O(N^2)$ porovnať so všetkými vrcholmi $O(N)$
- zlepšenie-heuristika-odstránenie vrcholov
napravo od skúmaného vrcholu $\rightarrow O(n^2)$



```

CreateGG(V)
{
    for (all p from V)
    {
         $N_p = V \setminus p$ ;
        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

```
CreatorSIG(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  $p_i p_j$ ;
        }
    }
}
```

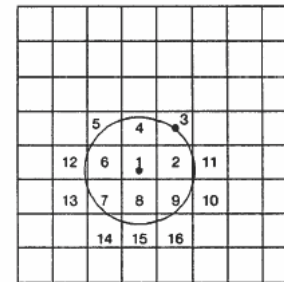


Fig. 1 Spiral nearest neighbor search using cells.

Voronoi diagram

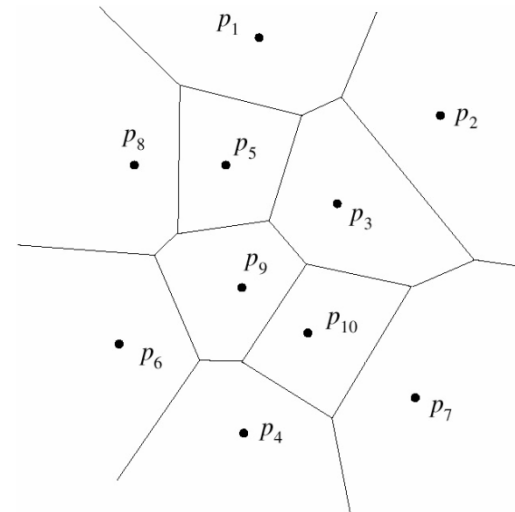
- geometrický graf
- pre danú množinu bodov $S = \{p_1, p_2, \dots, p_n\}$, každá stena VD je priradená jednému bodu p_i a pre každý bod oblasti je tento bod bližšie k p_i ako ku inému bodu z S

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

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

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

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

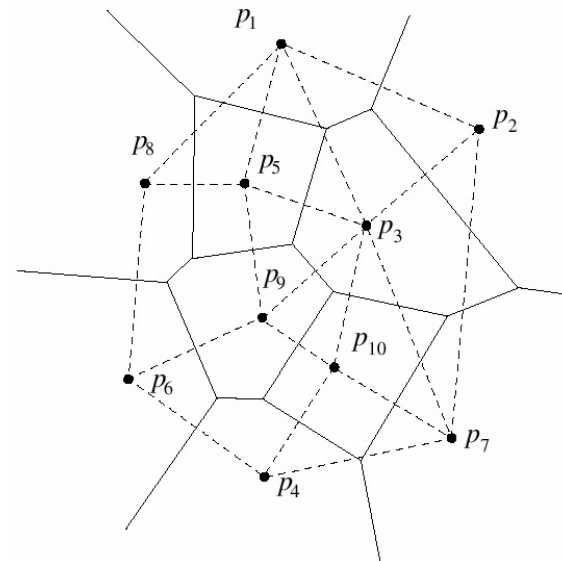


Vlastnosti VD

- $\text{VoR}(p_i, S)$ je prienikom max $n-1$ polrovín, je to otvorená a konvexná množina
- $\text{VoR}(p_i, S)$ je neohraničná práve vtedy keď p_i patrí do konvexného obalu S
- VD má $O(n)$ hrán a vrcholov
- priemerný počet hrán na hranici $\text{VoR}(p_i, S)$ je menší ako 6

Delaunayova Triangulácia

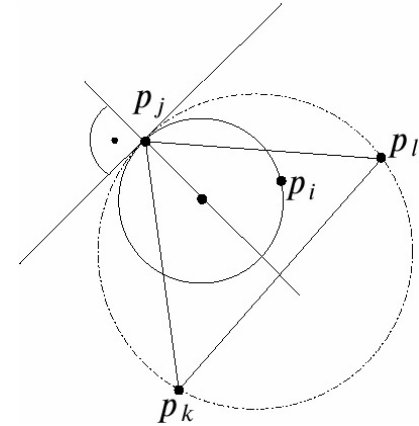
- pre množinu bodov S a $VD(S)$, $DT(S)$ je duálny graf k VD
- body p_i, p_j, p_k tvoria trojuholník v $DT \Leftrightarrow$ kružnica prechádzajúca tromi bodmi neobsahuje vnútri ani na hranici iný bod z S
- DT maximalizuje minimálny vnútorný uhol



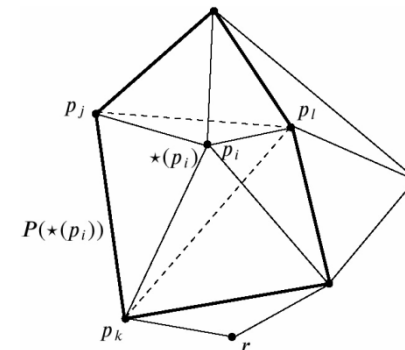
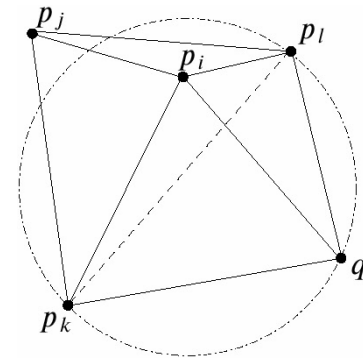
Konštrukcia DT

- 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 sa vykonáva otočenie hrany v „zlom“ štvoruholníku

1. prípad

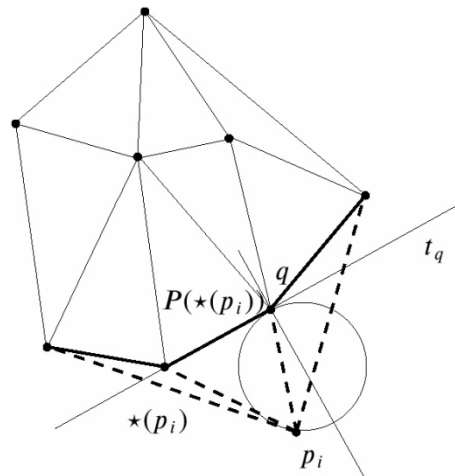


- p_i leží v $T = \Delta(p_j, p_k, p_l)$
- hrany $p_i p_j, p_i p_k, p_i p_l$ patria novej DT
- konflikt môže nastať v Δ ktoré sú susedné s T
- vtedy sa otočí hrana T
- konflikt aj v ďalších Δ
- ak hrany $P^*(p_i)$ sú z DT, koniec



2. prípad

- ak p_i leží z vonku konvexného obalu pôvodnej DT
- pre všetky body q z S , ktoré sú „viditeľné“ z p_i , $p_i q$ bude patriť novej DT
- niektoré hrany na hranici bude treba otočiť



Algoritmus

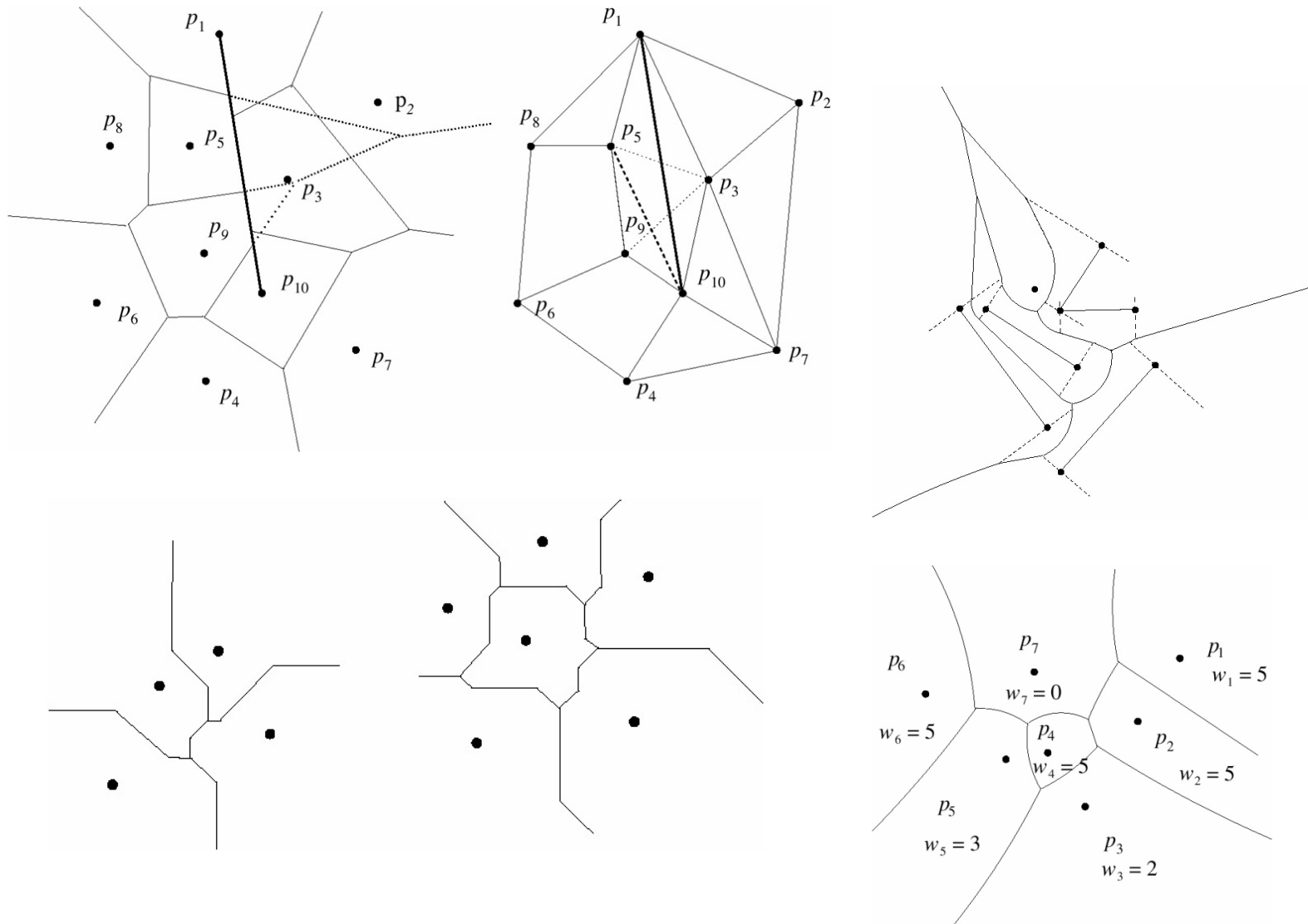
```
Delaunay(S) (S is set of sites)
{
    T = new array;
    while (S.size() > 0)
    {
        p = S.First;
        S.DeleteFirst;
        T.InsertSite(p);
    }
}
```

```
InsertSite(T, p) (T represents the current Delaunay triangulation, p is a new site)
{
    t = T.FindTriangle(p);
    if (t == NULL)
        Star(p) = t.OuterTriangle(p);
    else
        Star(p) := T.Edges(t, 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

Zovšeobecnenia



Mračná bodov

- jednoduchá množina bodov
- výstup z mnohých meracích zariadení - skenery
- rozvíjajúca sa oblasť v modelovaní a vizualizácii
- potreba definovania povrchu



Definovanie povrchu

- z množiny bodov implicitný povrch
- weighted least squares (WLS) interpolation
- povrch S je určený funkciou $f(\mathbf{x})$

$$S = \{\mathbf{x} \mid f(\mathbf{x}) = 0\}$$

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

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

vážený priemer
všetkých bodov

Váhovacie funkcie

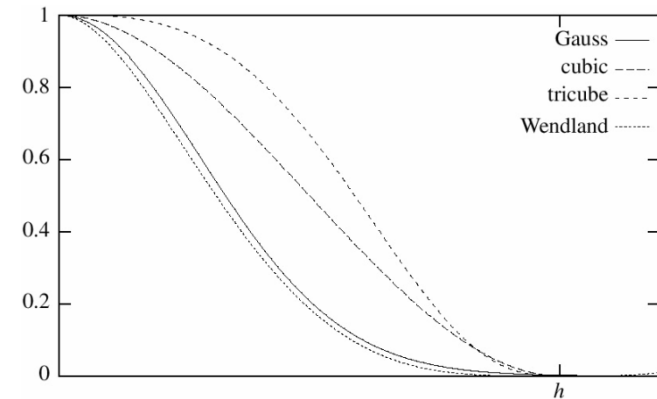
- Určujú váhu bodu podľa vzdialenosti

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

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



Parametre

- normála

- minimalizácia $\sum_{i=1}^N (\mathbf{n}(\mathbf{x}) \cdot (\mathbf{a}(\mathbf{x}) - \mathbf{p}_i))^2 \theta(\|\mathbf{x} - \mathbf{p}_i\|)$

- najmenší vlastný vektor matice $\mathbf{B} = (b_{ij})$

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

- vplyv vzdialenosti h

- detaily vs. vyhladenie

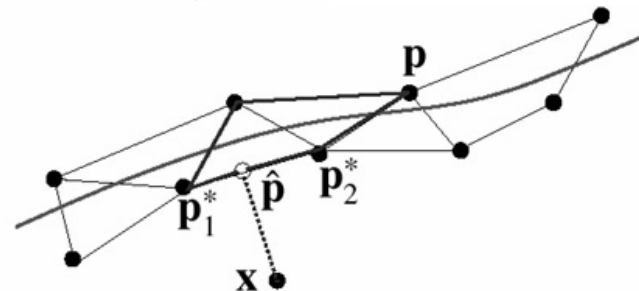
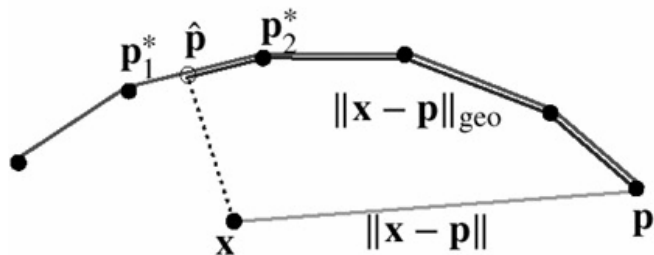
- \rightarrow adaptívne h

- použitie geodetickej vzdialenosti
 - určenie lokálnej hustoty vzoriek

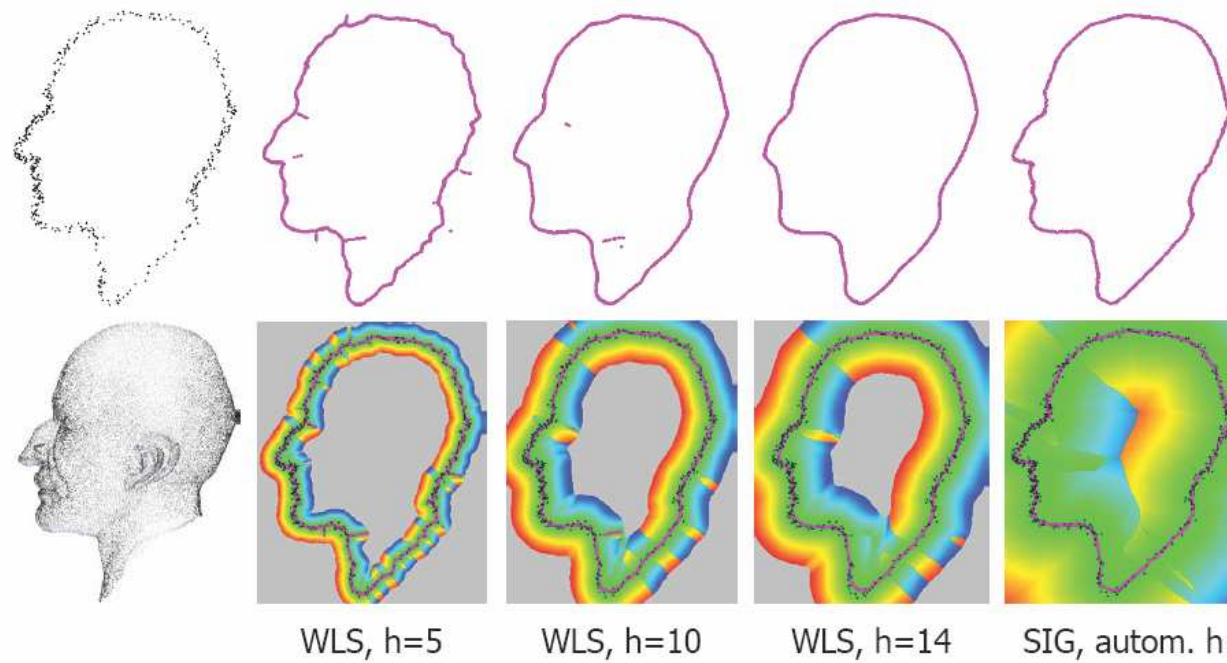
Využitie grafov

- problémy v zle vzorkovaných oblastiach s Euklid. vzdialenosťou - nespojitosti
- geodetická vzdialenosť na základe aproximácie povrchu grafom geometrickej susednosti (SIG...)
- $a = \| \hat{p} - p_1^* \|$

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



Výsledky



koniec (-:

`florek@sccg.sk`