

# Konvexní obálka množiny bodů.

Graham Scan. Jarvis Scan. Quick Hull. Minimum Area Enclosing Rectangle. Výpočet hlavních směrů budov. Generalizace budov LOD0.

Tomáš Bayer | bayertom@natur.cuni.cz

Katedra aplikované geoinformatiky a kartografie. Přírodovědecká fakulta UK.

# Obsah přednášky

- 1 Úvod do problému
- 2 Formulace problému
- 3 Metody konstrukce konvexní obálky
- 4 Minimum Area Enclosing Rectangle
- 5 Aplikace v kartografické generalizaci
- 6 Náhrada obdélníkem se stejnou plochou

# 1. Generalizace obrysu budovy



## 2. Formulace problému

Dáno: Množina  $n$  bodů  $P = \{p_1, p_2, \dots, p_n\}$  v  $\mathbb{R}^2$ , kde  $p_i = [x_i, y_i]$ .

Hledáme: Nejmenší konvexní množinu  $\mathcal{H}$ ,  $\forall p_i \in C$ . (Convex Hull)

### Definice 1:

Konvexní obálka množiny  $\mathcal{H}(P)$  konečné množiny  $P$  v  $\mathbb{E}^2$  představuje nejmenší konvexní mnohoúhelník  $S$  obsahující  $P$  (tj. neexistuje  $S' \subset S$  splňující definici).

### Definice 2:

Konvexní obálka  $\mathcal{H}$  množiny konečné množiny  $P$  představuje konvexní mnohoúhelník  $S$  s nejmenší plochou.

### Definice 3:

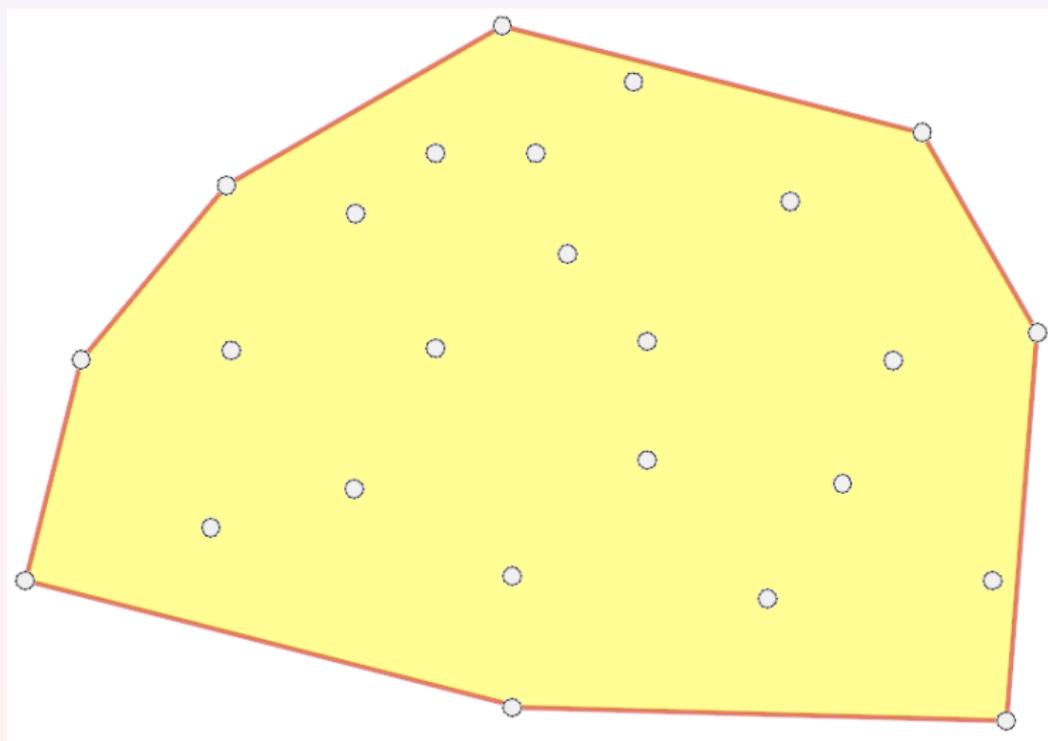
Konvexní obálka  $\mathcal{H}$  množiny konečné množiny  $P$  představuje průsečnici všech polorovin obsahujících  $P$ .

### Definice 4:

Konvexní obálka  $\mathcal{H}$  množiny konečné množiny  $P$  představuje sjednocení všech trojúhelníků, jejichž vrcholy tvoří body v  $P$ .

Množinu  $S$  označíme jako konvexní, pokud spojnice libovolných dvou prvků leží zcela uvnitř této množiny.

### 3. Ukázka konvexní obálky $\mathcal{H}$ množiny $P$



## 4. Využití konvexní obálky

Jedna z nejpoužívanějších geometrických struktur, pomocná struktura pro řadu algoritmů.

Často používána jako první odhad tvaru prostorového jevu

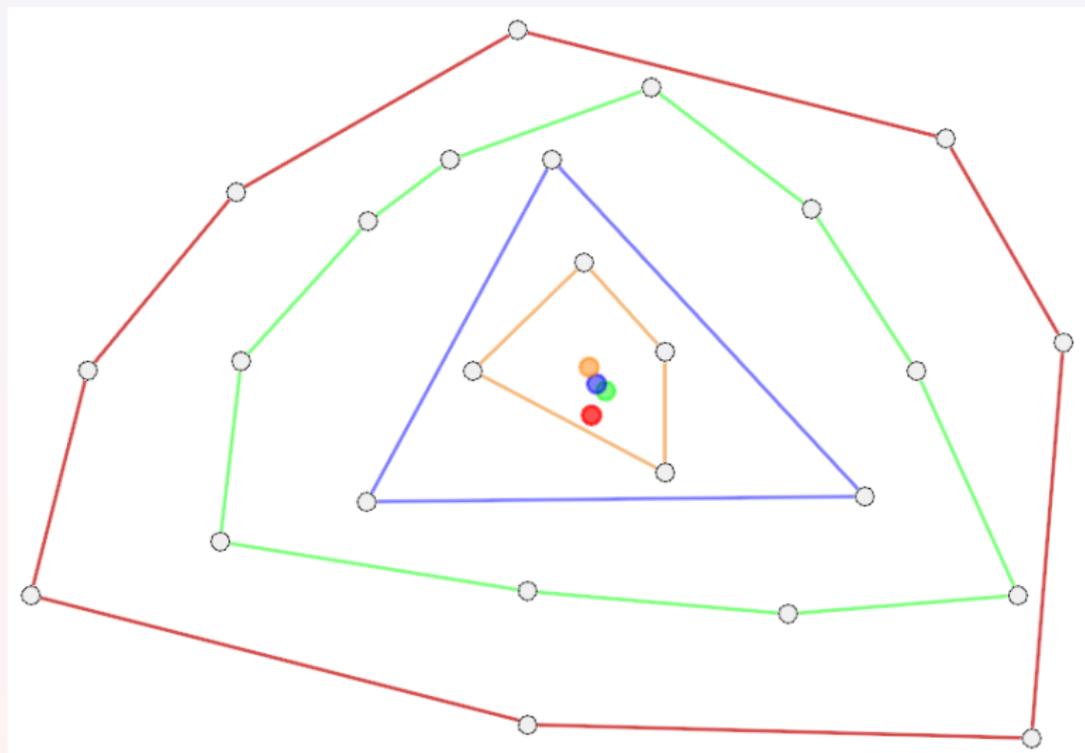
Příklady použití:

- Detekce kolizí: plánování pohybu robotů.
- Konstrukce MBR: využití v kartografii pro detekci tvaru a natočení budov.
- Analýza tvarů (Shape Analysis): analýza tvarů objektů.
- Statistická analýza: analýza rozptylů, odhady, např. metoda Onion Peeling.
- Analýza shluků (Cluster Analysis): vlastnosti clusterovaných dat.

Poznámka:

Konvexní obálka  $\mathcal{H}$  existuje v  $\mathbb{R}^d$ , budeme se zabývat pouze  $\mathbb{R}^2$  variantou.

## 5. Odhad centroidu metodou Onion Peeling



## 6. Metody konstrukce konvexní obálky

Přehled nejčastěji používaných metod pro konstrukci konvexní obálky:

- Jarvis Scan (Gift Wrapping).
- Graham Scan.
- Quick Hull.
- Inkrementální konstrukce.
- Zametací přímka.
- Divide and Conquer.

Některé z metod použitelné pouze pro  $\mathbb{R}^2$ , jiné převeditelné do vyšší dimenze.

## 7. Jarvis Scan (Jarvis, 1973)

Připomíná postup balení dárku do papíru (Gift Wrapping Algorithm).  
Jednoduchá implementace, lze rozšířit i do  $\mathbb{R}^3$ .

Předpoklad: v  $P$  nejsou tři kolineární body.

Nutnost předzpracování  $O(n)$ : nalezení pivotu  $q$

$$q = \min_{\forall p_i \in P} (y_i).$$

### Princip algoritmu:

Dva poslední body  $\mathcal{H}$  označeny  $p_{j-1}, p_j$ .

Aktuálně přidávaný bod  $p_{j+1}$  do  $\mathcal{H}$  maximalizuje úhel

$$p_{j+1} = \arg \max_{\forall p_i \in P} \angle(p_{j-1}, p_j, p_i).$$

Bod  $p_{j+1}$  hledáme ze všech  $p_i \in P$ , které dosud nejsou součástí  $\mathcal{H}$ .

Lze je značkovat.

## 8. Algoritmus Jarvis Scan

Nevýhodou složitost  $O(n^2)$ , nehodí se pro velké  $P$ .  
Snadná implementace.

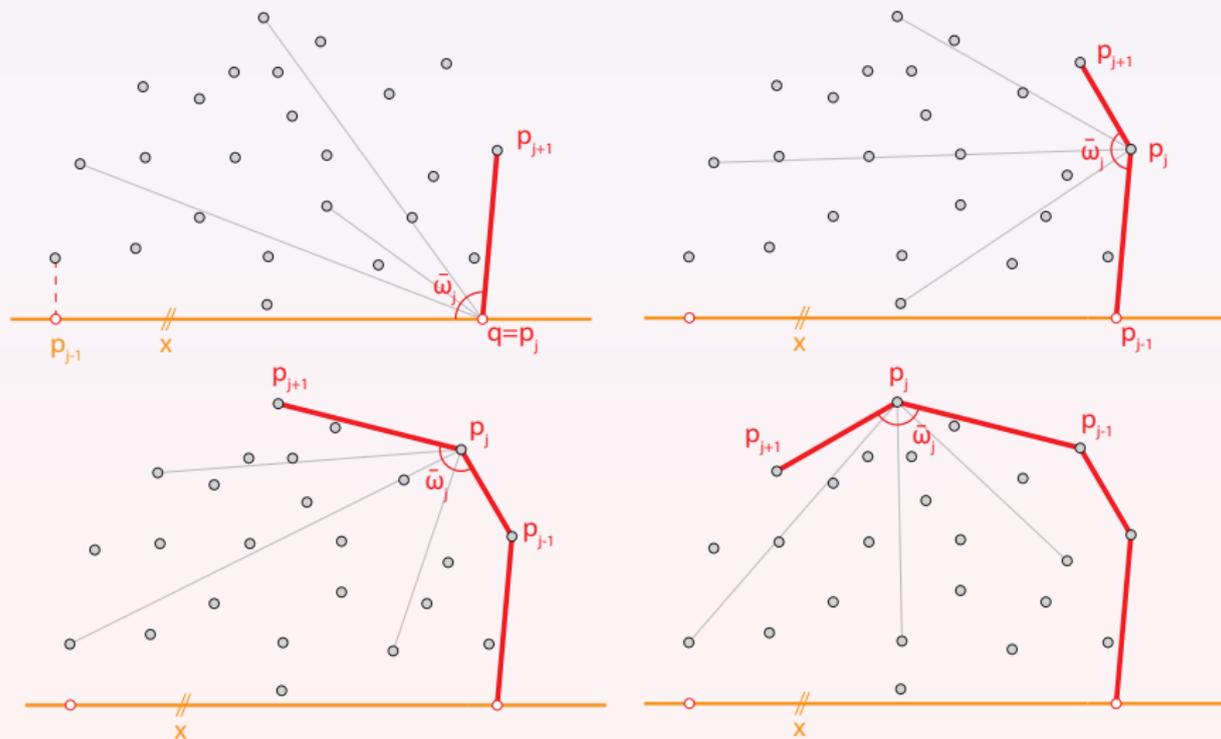
---

### Algoritmus 1: Gift Wrapping ( $P, \mathcal{H}$ )

---

- 1: Nalezení pivota  $q, q = \min(y_i)$ ,
  - 2: Přidej  $q \rightarrow \mathcal{H}$ .
  - 3: Inicializuj:  $p_{j-1} \in X, p_j = q, p_{j+1} = p_{j-1}$ .
  - 4: Opakuj, dokud  $p_{j+1} \neq q$ :
  - 5:       Nalezni  $p_{j+1} = \arg \max_{\forall p_i \in P} \angle(p_{j-1}, p_j, p_i)$
  - 6:       Přidej  $p_{j+1} \rightarrow \mathcal{H}$ .
  - 7:        $p_{j-1} = p_j; p_j = p_{j+1}$ .
-

## 9. Ukázka Jarvis Scan



## 10. Graham Scan

Algoritmus není možné rozšířit do  $\mathbb{R}^3$  (Graham 1972).

Časová složitost  $O(n \cdot \log(n))$ .

Lze použít i na rozsáhlé datasety.

Převod star-shaped polygonu na  $\mathcal{H}$ .

### Princip:

Konstrukce star-shaped polygonu z pivotu  $q$ .

Tvoření vrcholů seřazenými dle úhlu  $\omega = \angle(x, qp_i)$ .

Body ve star-shaped polygonu: index  $j$ .

Kritérium levotočivosti

$$p_j \begin{cases} \notin \mathcal{H}, & p_{j+1} \in \sigma_r(p_j, p_{j+1}), \\ ? \in \mathcal{H}, & p_{j+1} \in \sigma_l(p_j, p_{j+1}). \end{cases}$$

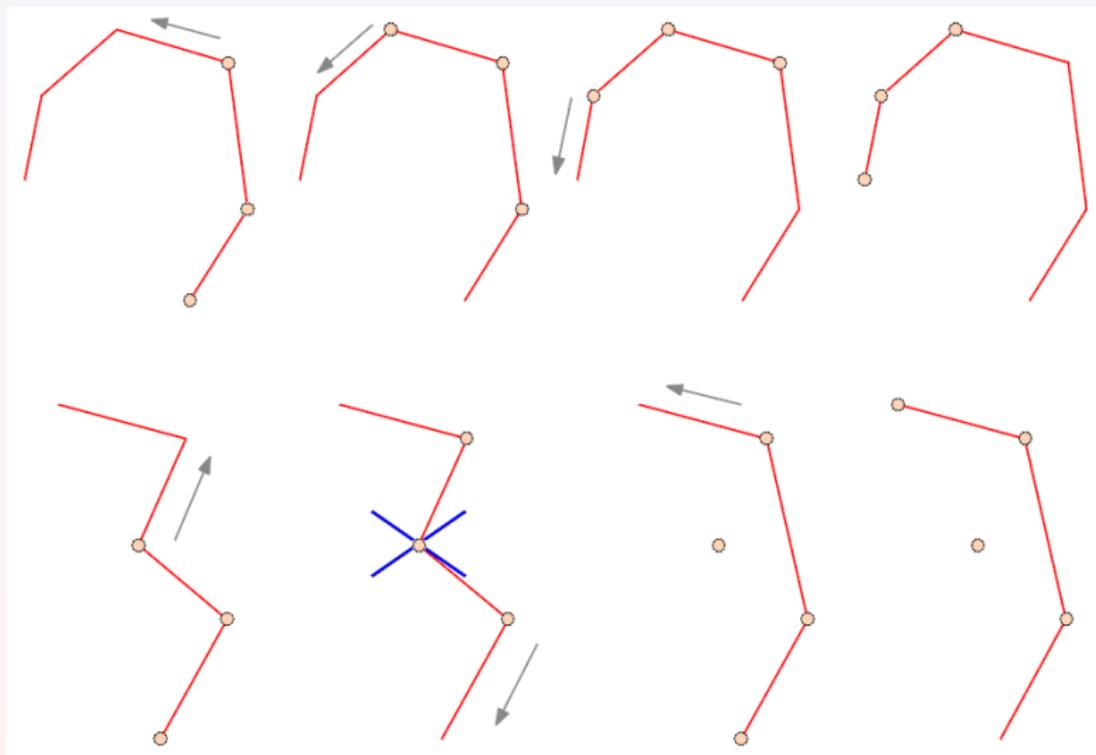
### Implementace:

Prvky v zásobníku  $S$ :  $t$  (top),  $t - 1$  (second).

Reflexní vrchol  $p_t$  odebrán z  $S$ , "konvexní"  $p_j$  přidán

$$S = \begin{cases} S.pop(), & p_j \in \sigma_r(p_{t-1}, p_t), \\ S.push(p_j), & p_j \in \sigma_l(p_{t-1}, p_t). \end{cases}$$

## 11. Ilustrace kritéria levotočivosti



## 12. Algoritmus Grahamova skenování

Předzpracování  $O(n)$ : pivot  $q$

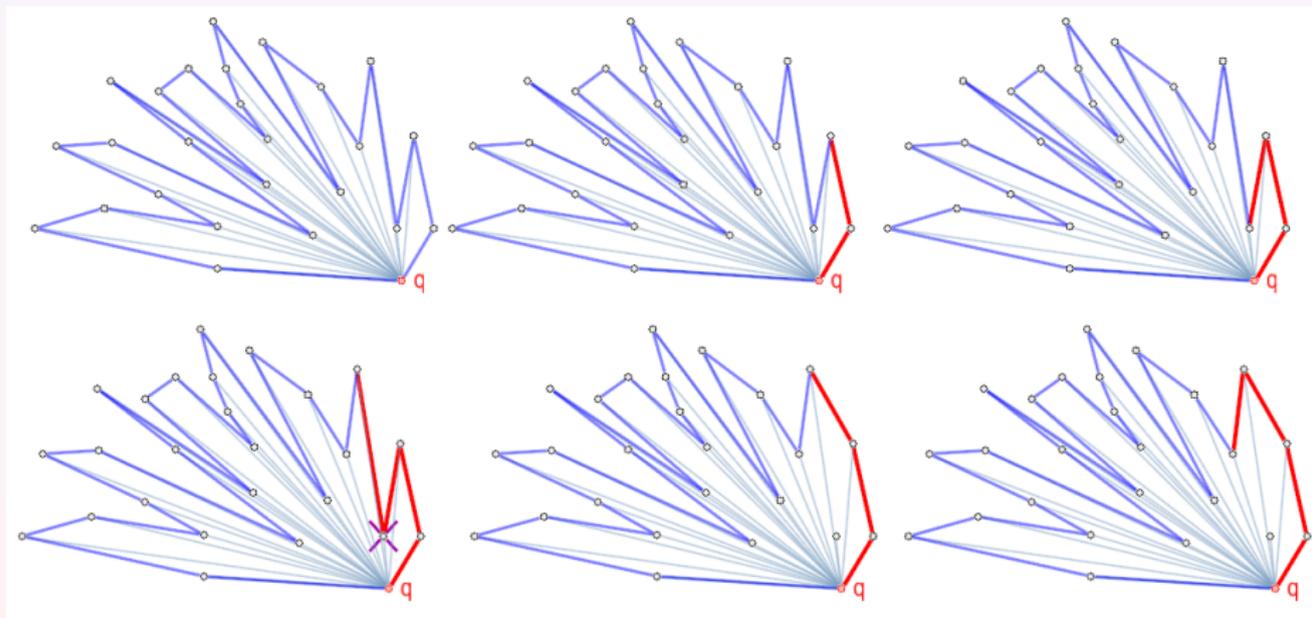
$$q = \min_{\forall p_i \in P} (y_i).$$

Předzpracování  $O(n \cdot \log(n))$ : setřídění  $P$  dle úhlu  $\angle(x, qp_i)$ .

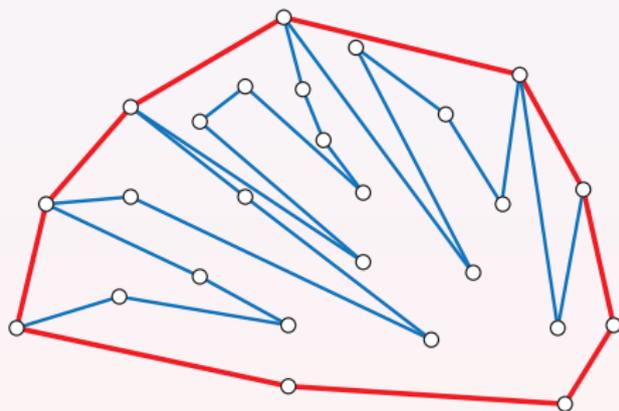
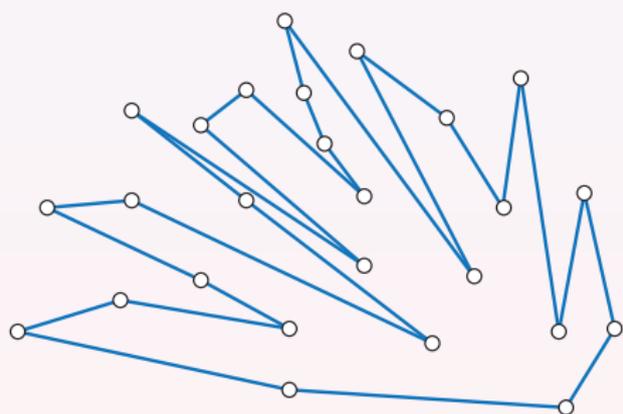
### Algoritmus 2: Graham Scan ( $P, S$ )

- 1: Nalezení pivota  $q = \min_{\forall p_i \in S} (y_i)$ ,  $q \in \mathcal{H}$ .
- 2: Setřídění  $\forall p_i \in S$  dle  $\omega_i = \angle(p_i, q, x)$ , index  $j$  odpovídá setříděnému pořadí.
- 3: Pokud  $\omega_k = \omega_l$ , vymaž bod  $p_k, p_l$  bližší ke  $q$ .
- 4: Inicializuj  $j = 2$ ;  $S = \emptyset$
- 5:  $S \leftarrow q$ ,  $S \leftarrow p_1$  (indexy posledních dvou prvků  $p_t, p_{t-1}$ )
- 5: Opakuj pro  $j < n$ :
  - 6:       if  $p_j$  vlevo od  $p_{t-1}, p_t$ :
  - 7:              $S \leftarrow p_j$
  - 8:              $j = j + 1$
  - 9:       else pop  $S$ .

# 13. Ukázka konstrukce Graham Scan



# 14. Ukázka konstrukce Graham Scan



# 15. Algoritmus Quick Hull

Strategie Divide & Conquer.

Ve většině případů rychlý  $\Theta(n \lg n)$ , avšak  $O(n^2)$ .

Obálka postupně expanduje všemi směry.

## Princip algoritmu:

Předzpracování: setřídění bodů dle  $x$ ,  $O(n \lg n)$ .

Rozdělení  $P$  na  $P_U$  (upper) a  $P_L$  (lower) vzhledem k  $(q_1, q_3)$

$$q_1 = \min_{\forall p_i \in P} (x_i), \quad q_3 = \max_{\forall p_i \in P} (x_i).$$

$\mathcal{H}$  konstruována ze 2 částí: *Upper Hull* ( $\mathcal{H}_U$ ), *Lower Hull* ( $\mathcal{H}_L$ ).

$\mathcal{H}_U \subset P_U$ ,  $\mathcal{H}_L \subset P_L$ .

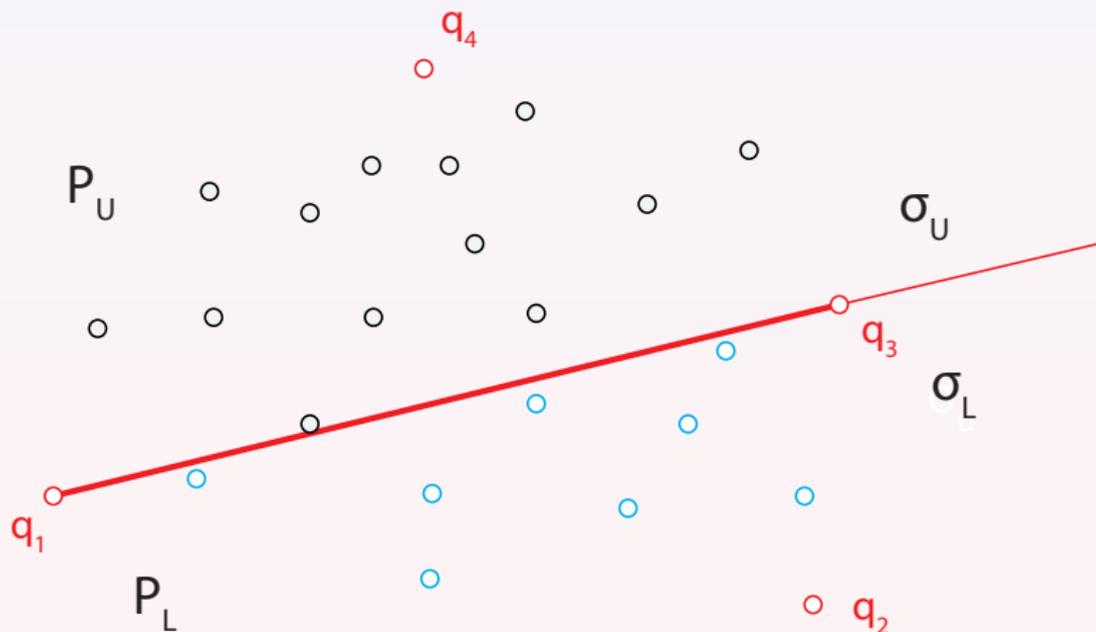
$\mathcal{H}_U$ ,  $\mathcal{H}_L$  zpracovány samostatně

$$\mathcal{H} = \mathcal{H}_U \cup \mathcal{H}_L.$$

## Důvod rychlosti:

Většinou málo rekurzivních kroků, málo bodů vně aproximace  $\mathcal{H}_U$ ,  $\mathcal{H}_L$ .

Pokud  $S \equiv \mathcal{H} \Rightarrow$  worst case.

16. Rozdělení na  $P_U, P_L$ 

# 17. Algoritmus Quick Hull, globální procedura

Nalezení  $q_1, q_2$ , nutné předzpracování.

- Rozdělení  $P$  na:  $P_U \subset P$  (upper) a  $P_L \subset P$  (lower) vzhledem k  $(q_1, q_2)$ .
- Přidání  $\mathcal{H}_U \leftarrow q_1, q_3$ , rekurzivní procedura nad  $P_U$ .
- Přidání  $\mathcal{H}_L \leftarrow q_3, q_4$ , rekurzivní procedura nad  $P_L$ .

Při CCW orientaci vede ke konstrukci  $\mathcal{H}$  bez nutnosti spojení.

---

Algoritmus 3: Quick Hull ( $P, \mathcal{H}$ ), globální procedura

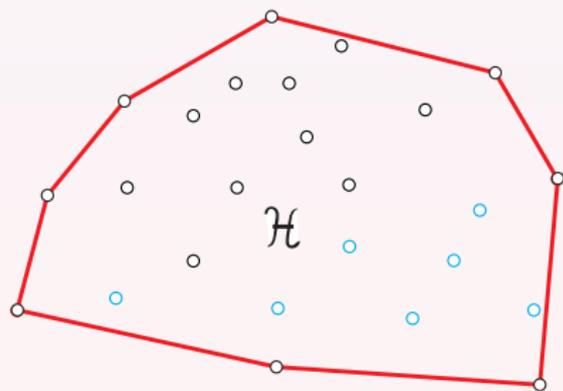
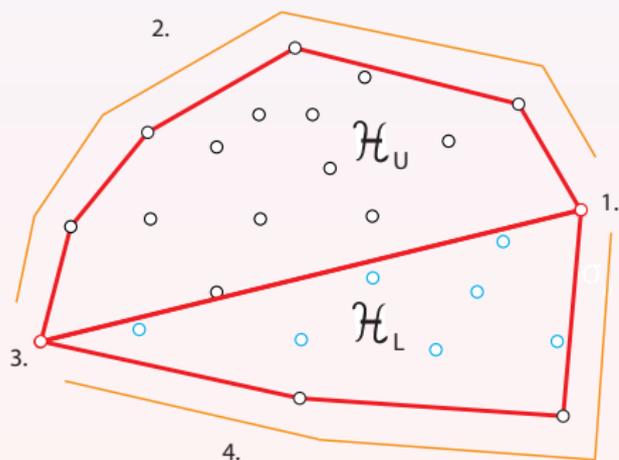
---

- 1:  $\mathcal{H} = \emptyset, P_U = \emptyset, P_L = \emptyset$
- 2:  $q_1 = \min_{\forall p_i \in S}(x_i), q_3 = \max_{\forall p_i \in S}(x_i)$ .
- 3:  $P_U \leftarrow q_1, P_U \leftarrow q_3$ .
- 4:  $P_L \leftarrow q_1, P_L \leftarrow q_3$ .
- 5: for  $\forall p_i \in P$
- 6:       if  $(p_i \in \sigma_l(q_1, q_3)) P_U \leftarrow p_i$
- 7:       else  $P_L \leftarrow p_i$
- 8:  $\mathcal{H} \leftarrow q_3$ . //Přidej bod  $c$  do  $\mathcal{H}$ .
- 9: Quick Hull(1, 0,  $P_U, \mathcal{H}$ ) //Upper Hull
- 10:  $\mathcal{H} \leftarrow q_1$ . //Přidej bod  $c$  do  $\mathcal{H}$ .
- 11: Quick Hull(0, 1,  $P_L, \mathcal{H}$ ) //Lower Hull

# 18. Spojení do $\mathcal{H}$

Vhodné načasování operací zajišťuje spojení  $\mathcal{H}_U$  a  $\mathcal{H}_L$  v globální proceduře:

- 1 Přidání  $q_3$ :  $\mathcal{H} \leftarrow q_3$ .
- 2 Rekurze  $\mathcal{H}_U$ .
- 3 Přidání  $q_1$ :  $\mathcal{H} \leftarrow q_1$ .
- 4 Rekurze  $\mathcal{H}_L$ .



# 19. Algoritmus Quick Hull, lokální procedura

Důležité pořadí operací (body  $\mathcal{H}$  CCW orientovány).

Vstupní hrana  $(p_s, p_e)$  aproximované  $\mathcal{H}$ .

1) Nalezení bodu  $\bar{p}$  s indexem  $\bar{i}$

$$\bar{p} = \arg \max_{\forall p_i \in S} \|p_i - (p_s, p_e)\|_2, \quad \bar{p} \in \sigma_r(p_s, p_e).$$

2) Vytvoření nových segmentů  $(p_s, \bar{p})$  a  $(\bar{p}, p_e)$ .

3) Rekurzivní volání nad prvním segmentem  $(p_s, \bar{p})$ .

4) Přidání  $\mathcal{H} \leftarrow \bar{p}$ .

5) Rekurzivní volání nad druhým segmentem  $(\bar{p}, p_e)$ .

V lokální proceduře předávány pouze indexy  $s, e$  koncových bodů segmentu.

Index  $\bar{i}$  odpovídá bodu  $\bar{p}$ .

Algoritmus 3: Quick Hull ( $s, e, P, \mathcal{H}$ ), lokální procedura

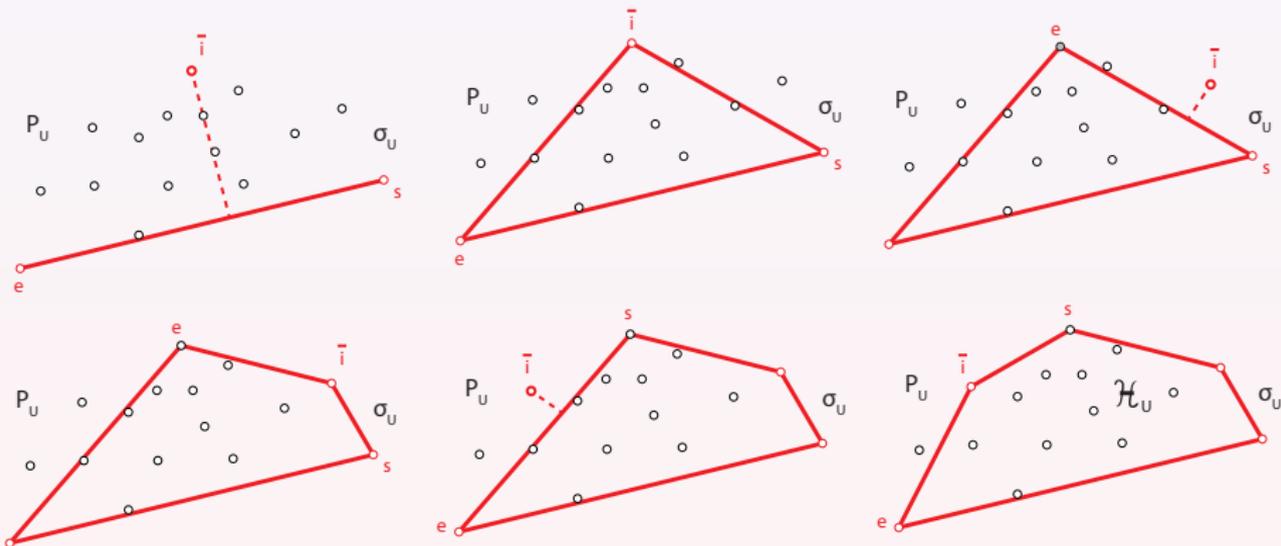
1: Najdi bod  $\bar{p} = \arg \max_{\forall p_i \in S} \|p_i - (p_s, p_e)\|, \bar{p} \in \sigma_r(p_s, p_e)$ .

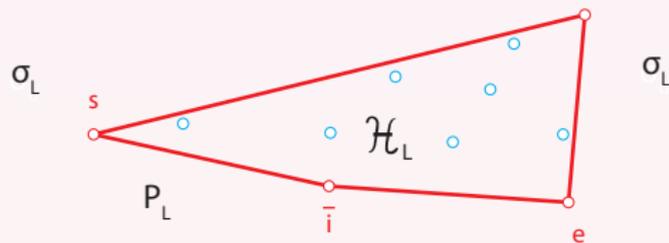
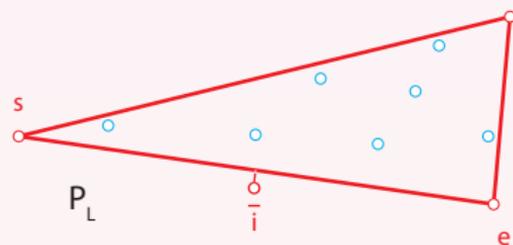
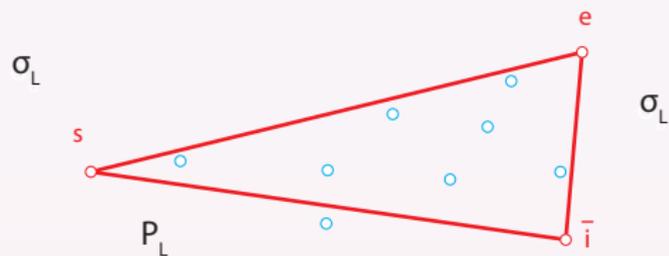
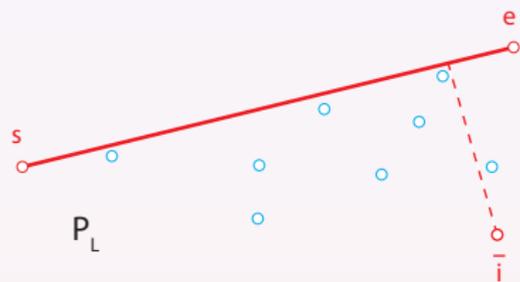
2: If  $\bar{p} \neq \emptyset$ ; //Existuje bod vpravo od hrany.

3: Quick Hull( $s, \bar{i}, P, \mathcal{H}$ ) //Upper Hull,  $\bar{i}$  index  $\bar{p}$

4:  $\mathcal{H} \leftarrow \bar{p}$ . //Přidej bod  $c$  do  $\mathcal{H}$ .

5: Quick Hull( $\bar{i}, e, P, \mathcal{H}$ ) //Lower Hull,  $\bar{i}$  index  $\bar{p}$

20. Lokální procedura  $P_U$ 

21. Lokální procedura  $P_L$ 

## 22. Doba běhu QHull

Časová funkce  $T(n)$ , očekávaná složitost (nejvzdálenější bod v mediánu):

$$\begin{aligned}
 T(n) &= 2T(n/2) + n, \\
 &= 2[2T(n/4) + n/2] + n = 4T(n/4) + 2n, \\
 &= 4[2T(n/8) + n/4] + 2n = 8T(n/8) + 3n, \\
 &= 8[2T(n/16) + n/8] + 3n = 16T(n/16) + 4n, \\
 &= iT(n/i) + n \log_2 i, \\
 &= n(T(1)) + n \log_2 n, \\
 &\leq n \log(n).
 \end{aligned}$$

Časová funkce  $T(n)$ , horní odhad složitosti (nejvzdálenější bod 2/n-1):

$$\begin{aligned}
 T(n) &= T(n-1) + n, \\
 &= [T(n-2) + n-1] + n = T(n-2) + n-1 + n, \\
 &= T(n-3) + n-2 + n-1 + n, \\
 &= T(n-i) + n-i+1 + \dots + n-2 + n-1 + n, \\
 &= T(0) + \frac{n}{2}(n+1), \\
 &\leq n^2.
 \end{aligned}$$

## 23. Minimum Area Enclosing Rectangle

Dáno: Množina  $n$  bodů  $P$  v  $\mathbb{R}^2$ .

Hledáme: Obdélník  $\mathcal{R}$  s nejmenší plochou opsaný  $S$ . (Minimum Area Enclosing/Bounding Rectangle).

Naivní algoritmy problém řeší v kvadratickém čase  $O(n^2)$ .

### Definice:

*Nejméně jedna strana obdélníka  $\mathcal{R}$  s minimální plochou  $A$  opsaného množině  $S$  je kolineární se stranou  $\mathcal{H}$ .*

Konstrukce MAER množiny  $P \Rightarrow$  předzpracování: konvexní obálka  $\mathcal{H}(P)$ .

Složitost konstrukce konvexní obálky  $\mathcal{H}(P)$ :  $O(n \cdot \log(n))$ .

Složitost konstrukce obdélníku  $\mathcal{R}(P)$  nad  $\mathcal{H}(P)$ : pouze  $O(n)$ .

Algoritmus řeší problém v čase  $O(n \cdot \log(n))$ .

Alternativa: Rotating Calipers (Toussaint, 1983), složitost  $O(n)$ , PCA (přibližně).

Využití v kartografii: kartografická generalizace (např. detekce natočení budov).

## 24. Princip konstrukce, opakovaná rotace

Body  $S$  s extrémními souřadnicemi:  $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ .

Snadno umíme sestavit MMB množiny  $P$ , který je  $\parallel$  s osami  $x, y$

$$V_1 = [\underline{x}, \underline{y}], \quad V_2 = [\bar{x}, \underline{y}], \quad V_3 = [\bar{x}, \bar{y}] \quad V_4 = [\underline{x}, \bar{y}].$$

Myšlenka:

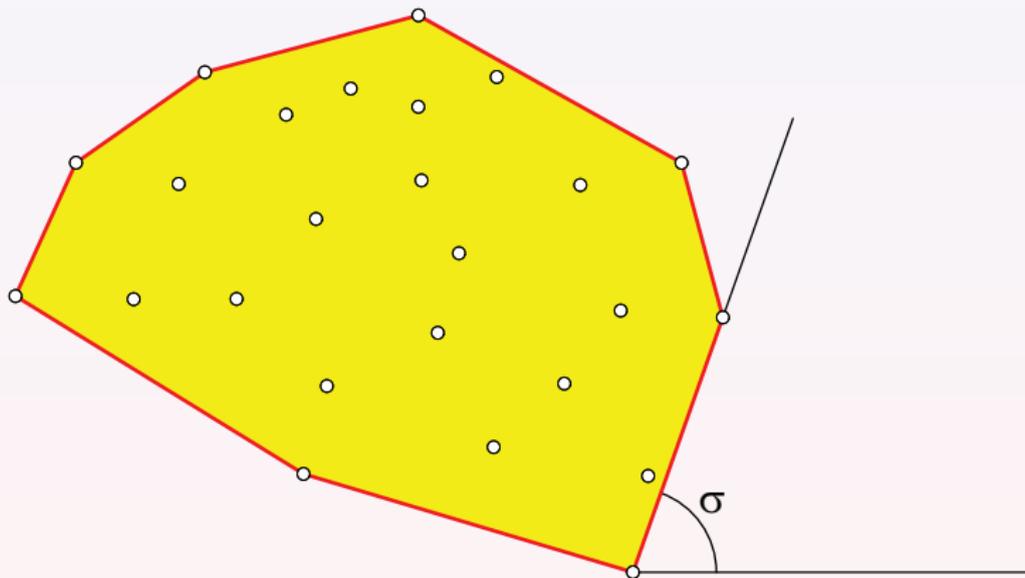
- Opakované otáčení  $P$  o úhel  $-\sigma$  (směrnice segmentu  $\mathcal{H}$ ).
- V této poloze nalezneme MMB množiny  $S$  a jeho plochu  $A$ .
- Otočením o  $\sigma$  min-max transformován na obecný obdélník ( $A$  se nezmění).
- Obdélník s minimální plochou bude  $\mathcal{R}$ .

---

### Algoritmus 6: Minimum Area Enclosing Rectangle

---

- 1: Najdi  $\mathcal{H} = CH(P)$
- 2: Inicializuj  $\mathcal{R} = MMB(P)$ ,  $\underline{A} = A(MMB(P))$
- 3: Opakuj pro každou hranu  $e$  obálky  $\mathcal{H}$ :
- 4:        Spočti směrnici  $\sigma$  hrany  $e$
- 5:        Otoč  $P$  o  $-\sigma$  :  $P_0 = R(-\sigma)P$ .
- 6:        Najdi  $MMB(P_0)$  a urči  $A(MMB(P_0))$ .
- 7:        Pokud  $A < \underline{A}$
- 8:                 $\underline{A} = A$ ,  $\underline{MMB} = MMB(P_0)$ ,  $\underline{\sigma} = \sigma$ .
- 9:  $\mathcal{R} = R(\underline{\sigma})\underline{MMB}$

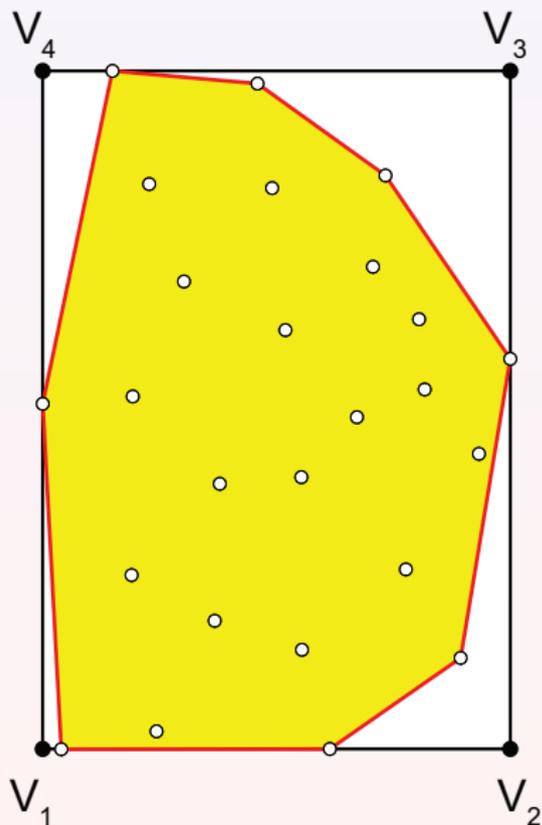
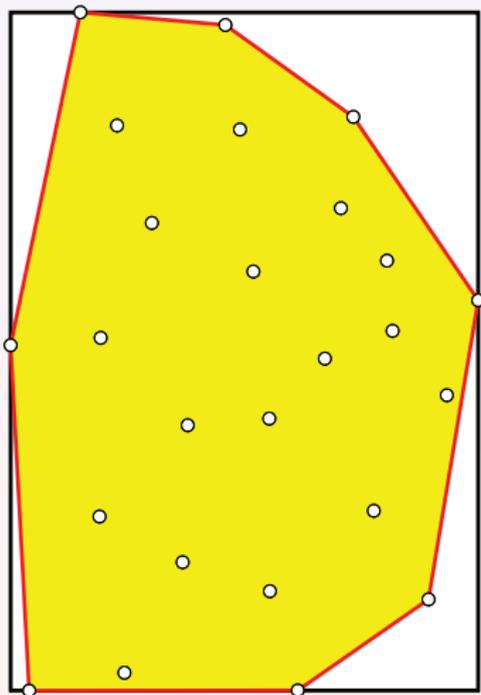
25. Rotace  $P$  o úhel  $-\sigma$ 

Rotace  $P$  o úhel  $-\sigma$

$$P_0 = R(-\sigma)P \Rightarrow \begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} \cos(-\sigma) & -\sin(-\sigma) \\ \sin(-\sigma) & \cos(-\sigma) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

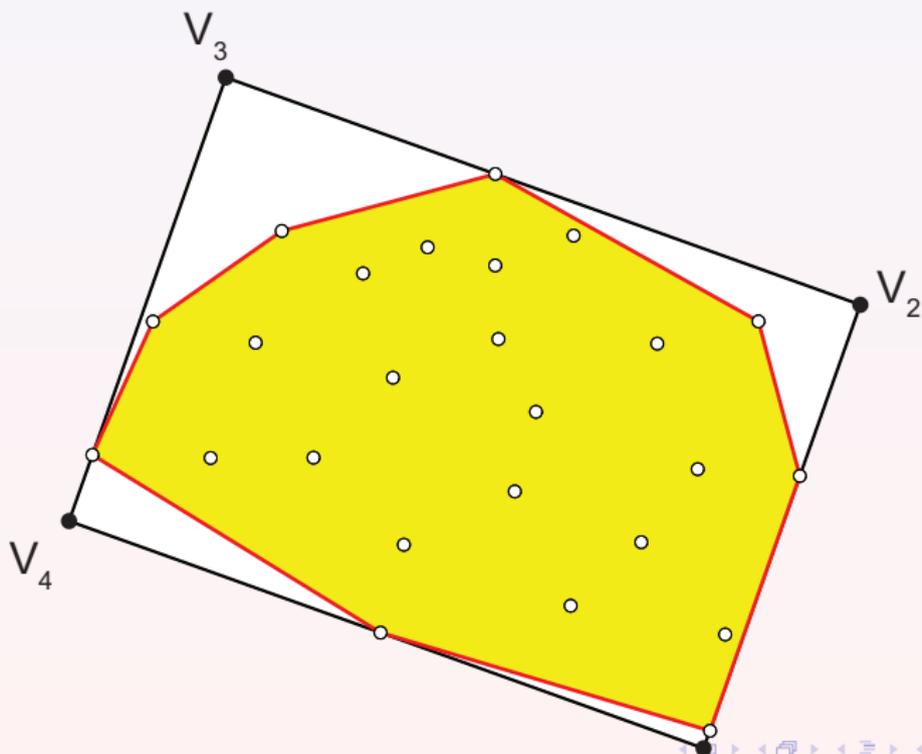
## 26. Ukázka, rotace $P$ o $-\sigma$

Rotace vrcholů  $P$  o  $-\sigma$ , nalezení extrémních bodů, konstrukce MMB.



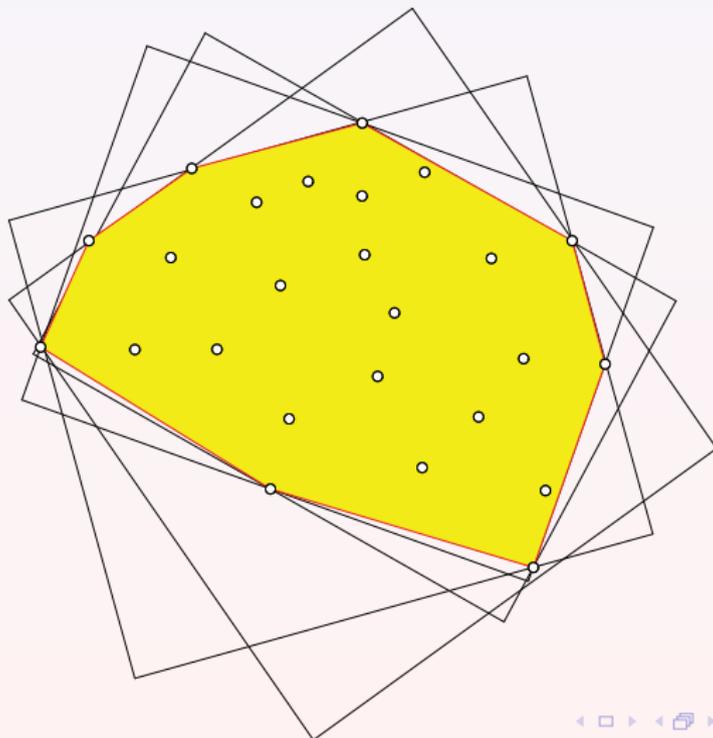
## 27. Zpětná rotace $P$ o $\sigma$

Plocha MMB se nezmění, výsledkem obecný obdélník.

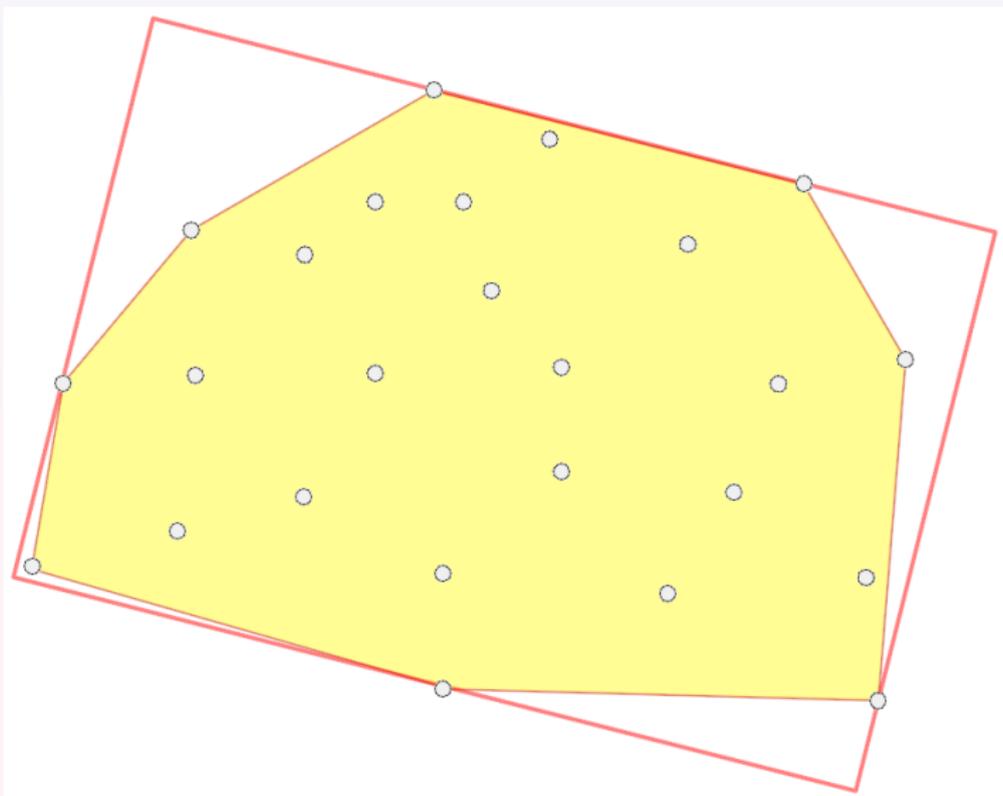


## 28. Postupné rotace

Opakované rotace  $-\sigma$ , konstrukce MMB, zpětné rotace o  $\sigma$ .  
Chyby ze zaokrouhlení.



## 29. Výsledek, obdélník s nejmenší plochou

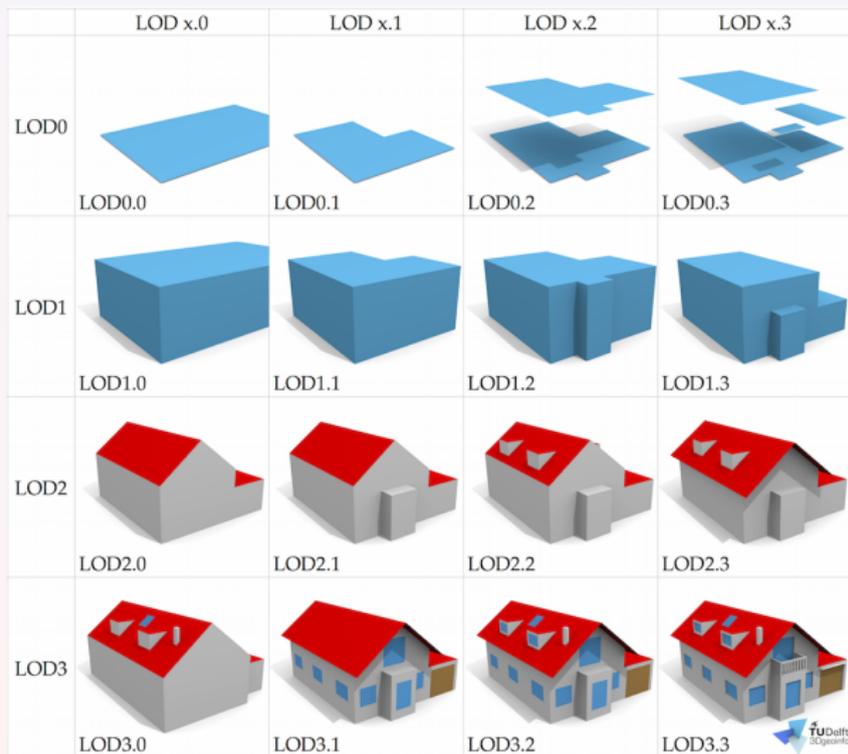


# 30. Reprezentace budov

Budovy lze reprezentovat s různými úrovněmi LOD.

Výsledek kartografické generalizace: 2D / 3D model.

Nutno detekovat hlavní směry budov + aproximace MBR.



## 31. Hlavní směry budovy

Častý problém při kartografické generalizaci budov či jejich automatickém rozpoznávání.

Budova před generalizací a po generalizaci musí mít uchovánu orientaci vzhledem k ostatním obsahovým prvkům mapy.

Např.: zachování uliční čáry.

Nutnost detekovat tzv. hlavní **směry budovy** (jsou na sebe zpravidla kolmé). Popisují orientaci (tj. natočení) budovy vzhledem k ostatním prvkům mapy.

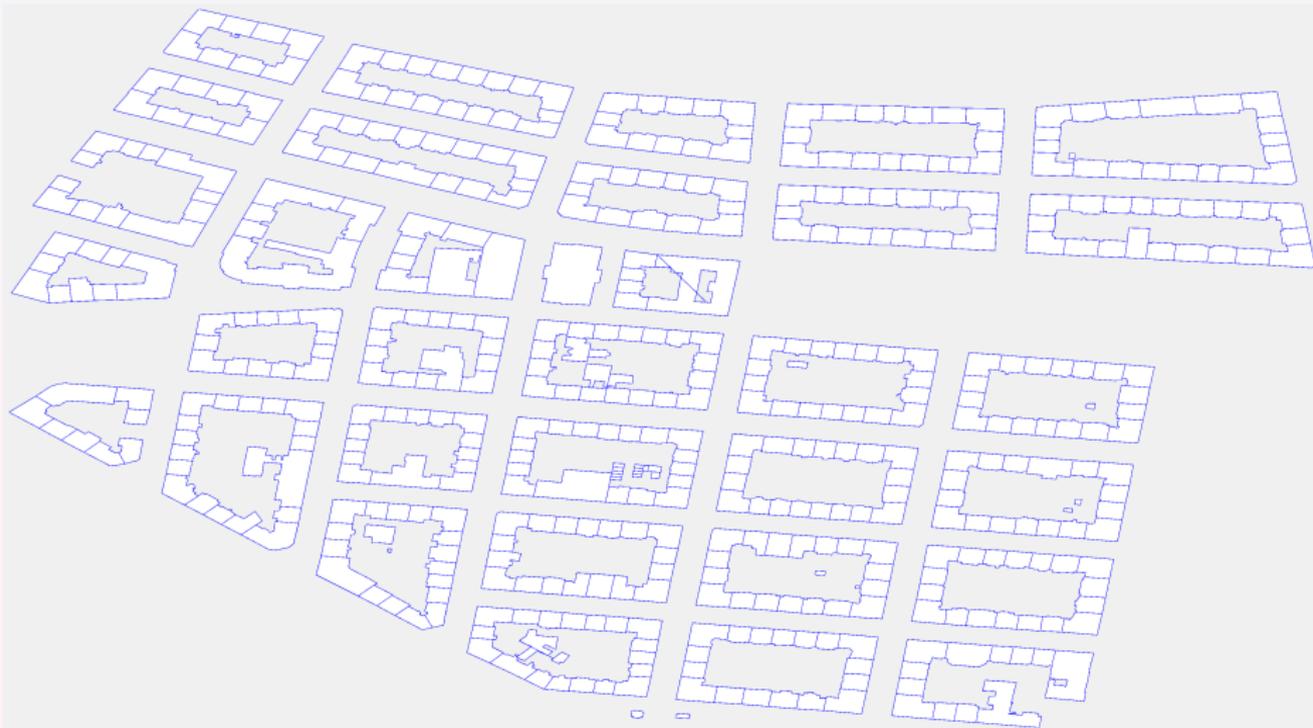
### Metody detekce natočení budov:

Založeny na aproximaci MAER (nějaký bounding box).

- PCA.
- Longest Edge.
- Weighted Bisektor.
- Minimum Area Enclosing Rectangle.
- Wall Average.

Strany bounding boxu definují hlavních směry.

## 32. Vstupní data



## 33. PCA

Pro nalezení hlavních směrů využívá singulární rozklad (SVD).

Snadno zobecnitelná do vyšších dimenzí.

Kovarianční matice

$$C = \begin{bmatrix} C(A, A) & C(A, B) \\ C(B, A) & C(B, B) \end{bmatrix}, \quad C(A, B) = \frac{1}{n-1} \sum_{i=1}^n (A_i - \mu_A)(B_i - \mu_B).$$

Singulární rozklad

$$C = U \Sigma V^T, \quad \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^T.$$

Matice  $U, V$ : vlastní vektory  $CC^T$  a  $C^T C$ , jednotkové (cos rotací  $\sigma$ )

$$U = V \equiv \begin{bmatrix} \cos \sigma & -\sin \sigma \\ \sin \sigma & \cos \sigma \end{bmatrix}.$$

Matice  $\Sigma$ : singulární hodnoty, čtverce vlastních čísel (velikosti vlastních vektorů)

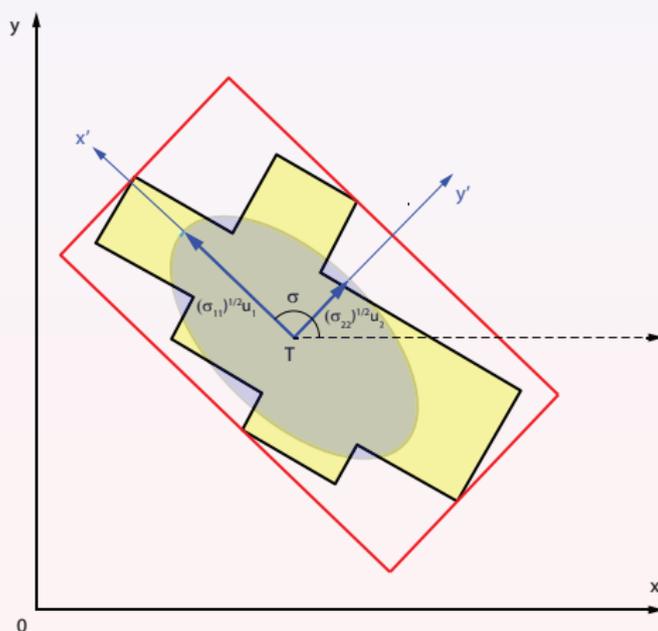
$$\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix}.$$

Rotace množiny  $P$  o úhel  $\pm\omega$

$$P_0 = PV, \quad P = V^{-1}P_0.$$

## 34. Ukázka PCA

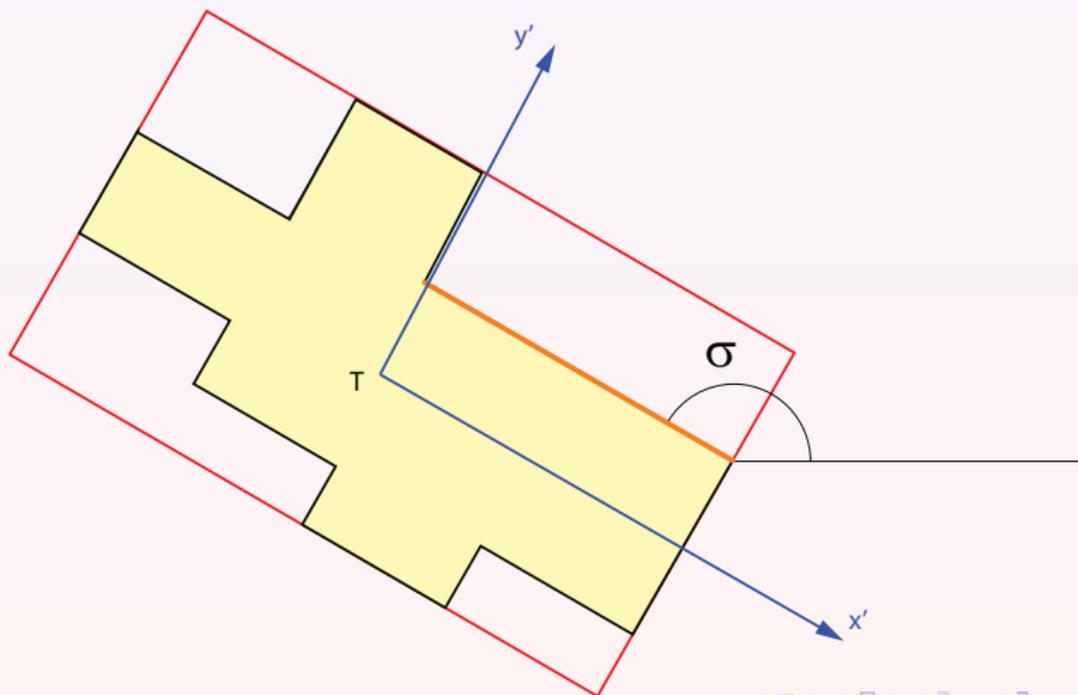
Hlavní směry budovy dány  $x'$ ,  $y'$ .



Pravidelné objekty (čtverec, kružnice): nízká efektivita (poměr plochy k MBR  $< 3.4$ ).

## 35. Longest Edge

První hlavní směr budovy představován nejdelší stranou v budově, druhý hlavní směr kolmý. Nedosahuje příliš dobrých výsledků. Nejdelší strana nemusí reprezentovat hlavní směr.



## 36. Longest Edge, výsledek



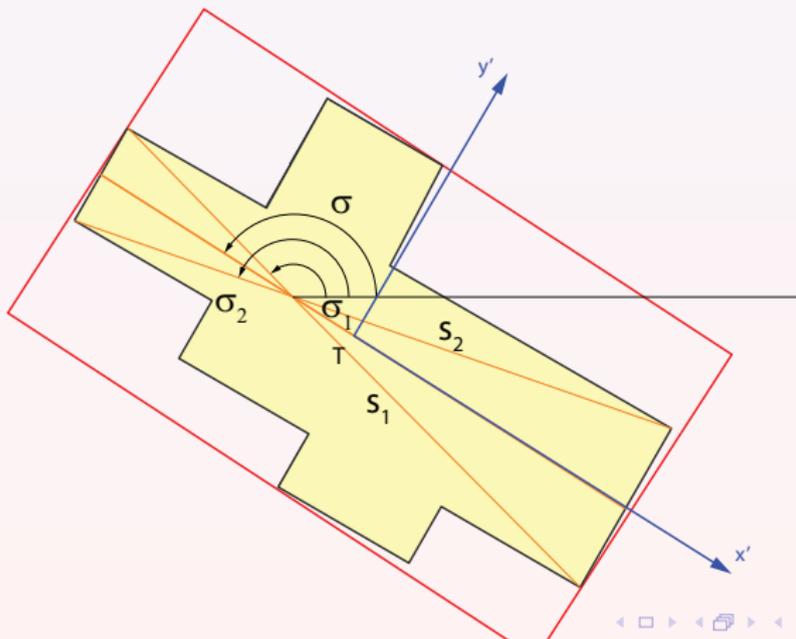
## 37. Weighted Bisector

Hledány dvě nejdelší úhlopříčky, směrnice  $\sigma_1, \sigma_2$ , délky  $s_1, s_2$ .

Hlavní směr dán váženým průměrem

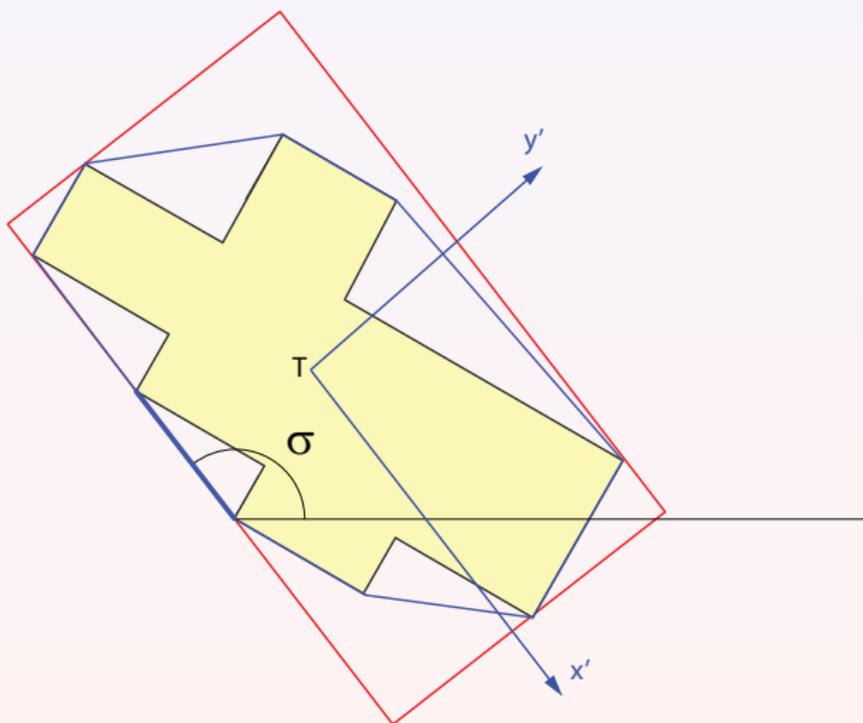
$$\sigma = \frac{s_1\sigma_1 + s_2\sigma_2}{s_1 + s_2}.$$

Dává velmi dobré výsledky.



## 38. Minimum Area Enclosing Rectangle

První hlavní směr představuje delší ze stran  $\mathcal{R}$ .  
Dává dobré výsledky, problémy s budovami tvaru L a Z.



## 39. Minimum Area Enclosing Rectangle, výsledek



## 40. Wall Average

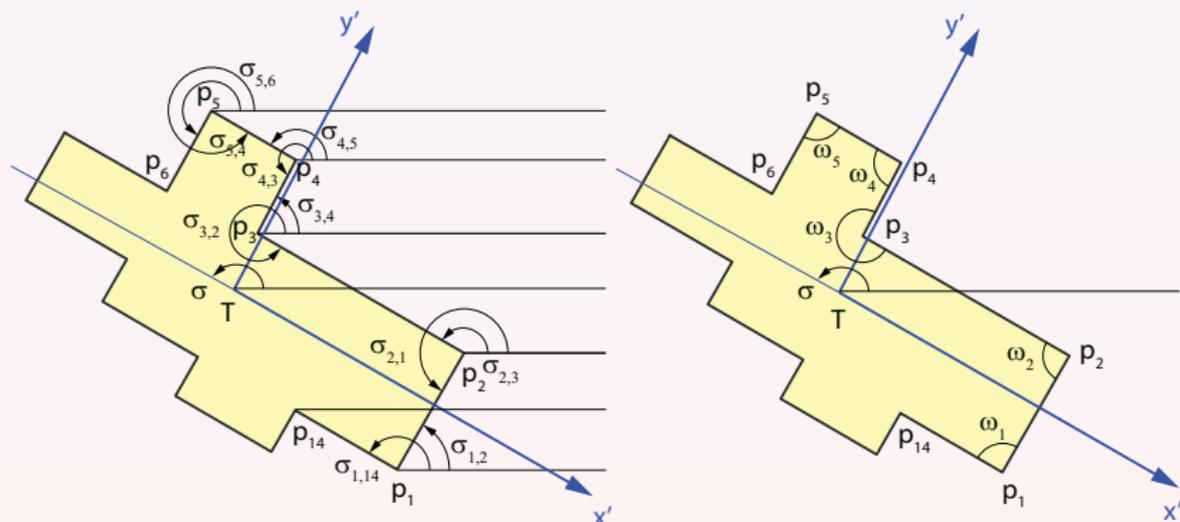
Na každou stranu budovy aplikována operace  $\text{mod}(\frac{\pi}{2})$ .

Ze “zbytků” hodnot spočten vážený průměr, váhou je délka strany.

Robustní metoda, avšak citlivá na “nepravé” úhly.

Nejprve určeny směrnice  $\sigma_i$  všech hran.

Poté spočteny vnitřní úhly  $\omega_j$ .



## 41. Wall Average: výpočet zbytku

Pro každý vrchol  $p_i$  spočteme vnitřní úhly

$$\omega_i = |\sigma_{i,i+1} - \sigma_{i,i-1}|.$$

Výpočet násobku  $\pi/2$

$$k_i = \frac{2\omega_i}{\pi}.$$

“Orientovaný” zbytek po dělení

$$r_i = (k_i - \lfloor k_i \rfloor) \frac{\pi}{2}.$$

Vnitřní úhel  $\omega_i \% \frac{\pi}{2} < \frac{\pi}{4}$  (např.  $92^\circ$ ):  $r_i > 0$ .

Vnitřní úhel  $\omega_i \% \frac{\pi}{2} > \frac{\pi}{4}$  (např.  $88^\circ$ ):  $r_i < 0$ .

Hlavní směr budovy

$$\sigma = \sigma_{1,2} + \sum_{i=1}^n \frac{r_i s_i}{S_i}.$$

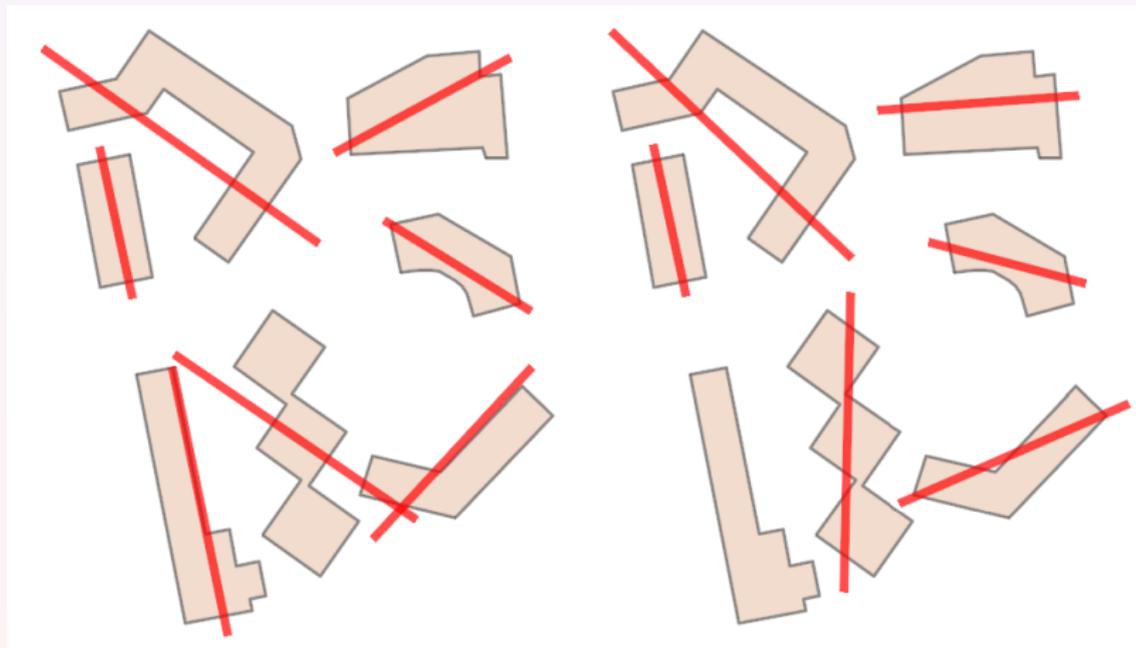
Vážený průměr zbytků po dělení.

## 42. Wall Average, výsledek



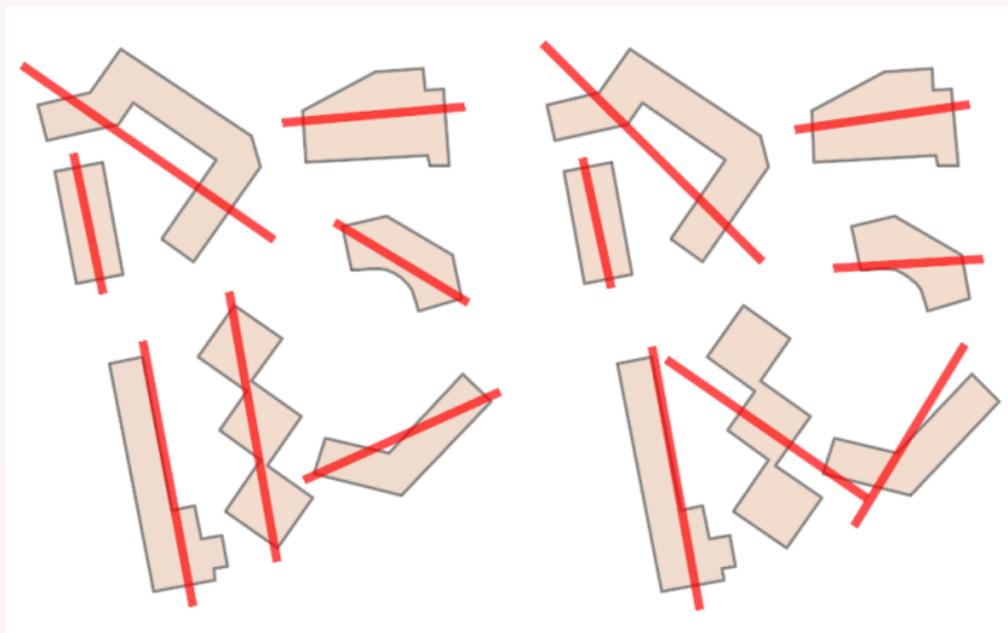
## 43. Detekce hlavních směrů budovy

Metody Longest Edge a Wall Average (Duchene et al, 2003).



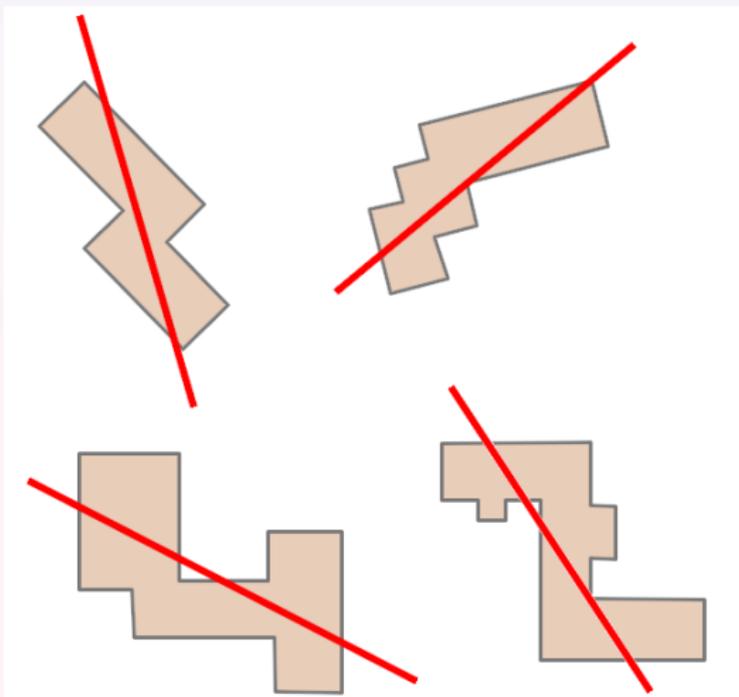
## 44. Detekce hlavních směrů budovy

Metody Minimum Area Enclosing Rectangle a Weigted Bisector (Duchene et al, 2003).



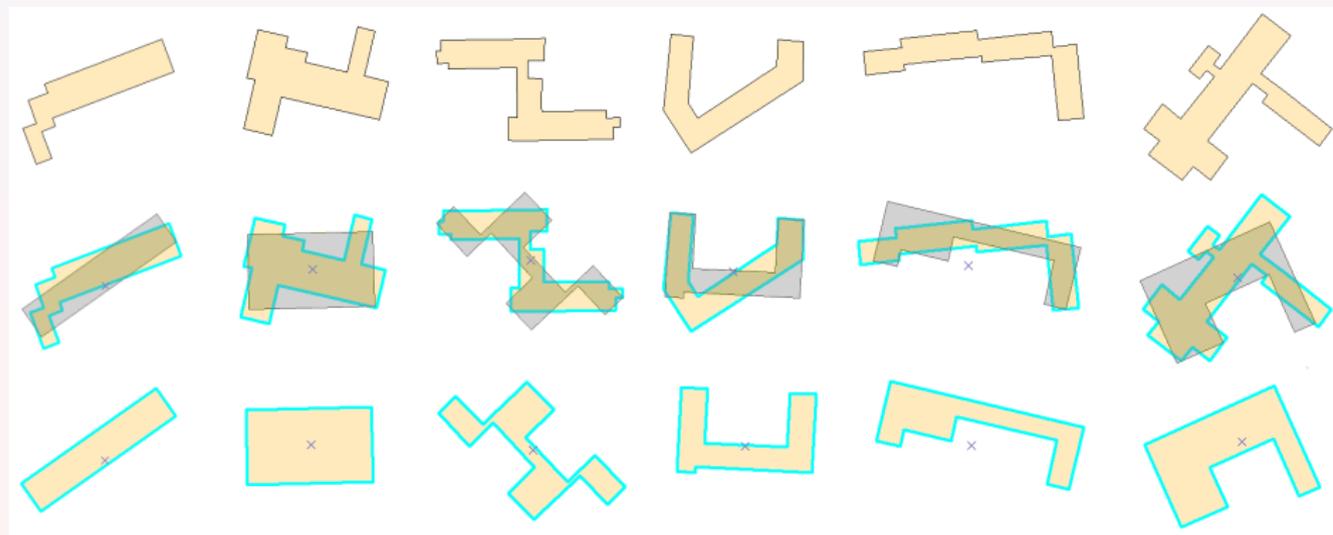
## 45. Problémy při detekci hlavních směrů

Problémy s detekcí hlavních směrů u budov tvarů L, Z.



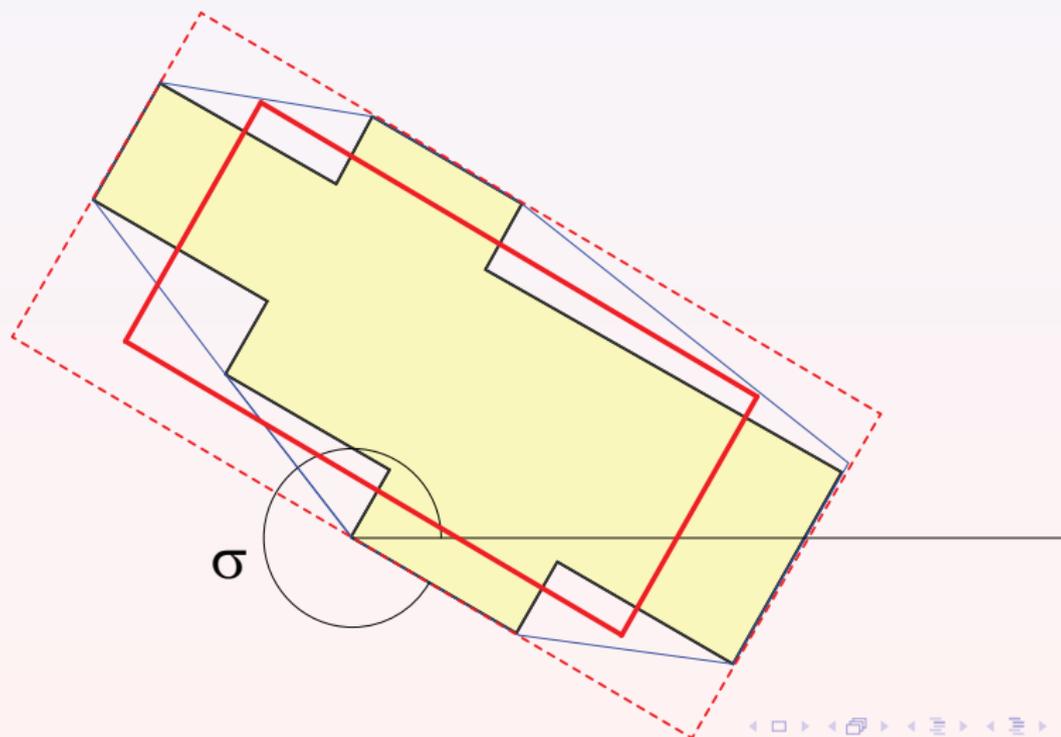
## 46. Vliv chybného určení hlavního směru na kartografickou generalizaci

Algoritmus Divide and Conquer (Bayer, 2009).



## 47. Náhrada obdélníkem se stejnou plochou

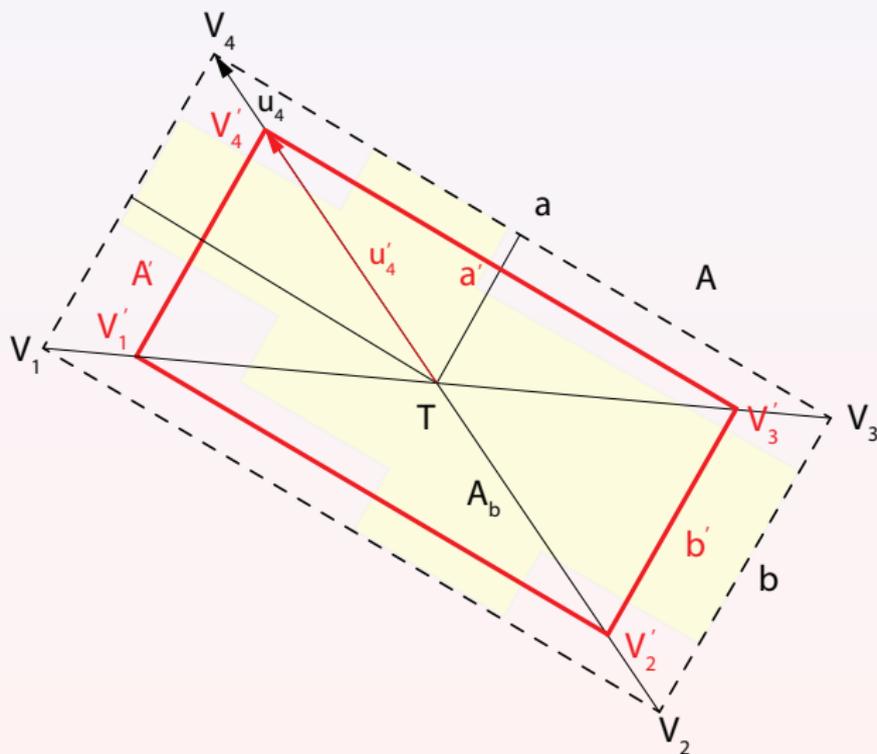
Často požadujeme, aby obdélník měl stejnou plochu jako generalizovaná budova.  
 Proporcionální zmenšení, společný střed  $T$ .



## 48. Ukázka

Obdélník  $R = (V_1, V_1, V_3, V_4)$ , strany  $a, b$ , plocha  $A$ .

Obdélník  $R' = (V'_1, V'_1, V'_3, V'_4)$ , strany  $a', b'$ , plocha  $A'$ .



## 49. Výpočet nových vrcholů obdélníku

Plochy obdélníků  $R, R'$

$$A = a \cdot b, \quad A' = kA = a' \cdot b' = \sqrt{k}a \cdot \sqrt{k}b,$$

kde

$$k = \frac{A_b}{A}.$$

Normy vektorů  $u_i$  a  $u'_i$  (polovina úhlopříčky) určíme z Pythagorovy věty

$$\|u_i\|_2^2 = \left(\frac{a}{2}\right)^2 + \left(\frac{b}{2}\right)^2, \quad \|u'_i\|_2^2 = k \left(\frac{a}{2}\right)^2 + k \left(\frac{b}{2}\right)^2 = k \|u_i\|_2^2.$$

Pak

$$\|u'_i\|_2 = \sqrt{k} \|u_i\|_2,$$

kde směrový vektor

$$u_i = V_i - T, \quad i = 1, \dots, 4.$$

Nový vrchol  $V'_i$  obdélníku

$$V'_i = T + u'_i = T + \sqrt{k}u_i.$$

## 50. Hodnocení efektivity detekce hlavních směrů

Zpravidla prováděno po jednotlivých objektech.

Předpoklad: objekty jsou pravoúhlé.

Hlavní směr  $\sigma$  (dán delší hranou MBR).

Směrnice jednotlivých segmentů  $\sigma_i$ .

Střední hodnota úhlových odchylek jednotlivých segmentů

$$\Delta\sigma_1 = \frac{\pi}{2n} \sum_{i=1}^n [r_i - r],$$

kde

$$r_i = (k_i - \lfloor k_i \rfloor) \frac{\pi}{2}, \quad k_i = \frac{2\sigma_i}{\pi}.$$

Střední hodnota čtverců úhlových odchylek jednotlivých segmentů

$$\Delta\sigma_2 = \frac{\pi}{2n} \sqrt{\sum_{i=1}^n (r_i - r)^2}.$$

Kartografická praxe: akceptace objektů  $\Delta\sigma < 10^\circ$ .