TECHNICKÁ ZPRÁVA

ÚLOHA Č.2: GENERALIZACE BUDOV LOD0

1. ZADÁNÍ:

Vstup: Množina budov $B = \{B_i\}_{i=1}^{n}$, budova $B_i = \{P_{i,j}\}_{j=1}^{m}$

Výstup: $G(B_i)$

Ze souboru načtete vstupní data představována lomovými body budov a proveďte generalizaci budov doúrovně detailu LOD0. Pro tyto ´účely použijte vhodnou datovou sadu, např. ZABAGED, testovaní proveďte nad třemi datovými sadami (historické centrum města, intravilán – sídliště, intravilán - izolovaná zástavba). Pro každou budovu určete její hlavní směry metodami:

- Minimum Area Enclosing Rectangle,
- PCA.

U první metody použijte některý z algoritmů pro konstrukci konvexní obálky. Budovu při generalizaci do úrovně LOD0 nahraďte obdélníkem orientovaným v obou hlavních směrech, se středem v těžišti budovy, jeho plocha bude stejná jako plocha budovy. Výsledky generalizace vhodně vizualizujte. Otestujte a porovnejte efektivitu obou metod s využitím hodnotících kritérií. Pokuste se rozhodnout, pro které tvary budov dávají metody nevhodné výsledky, a pro které naopak poskytují vhodnou aproximaci.

SKUPINA: Kateřina Chromá, Štěpán Šedivý

2. ZPRACOVANÉ DOBROVOLNÉ ČASTI ÚKOLU:

- *Generalizace budov metodou Longest Edge.* *5b*
- *Generalizace budov metodou Wall Average.* *8b*
- *Generalizace budov metodou Weighted Bisector.* *5b*

3. VSTUPNÍ DATA:

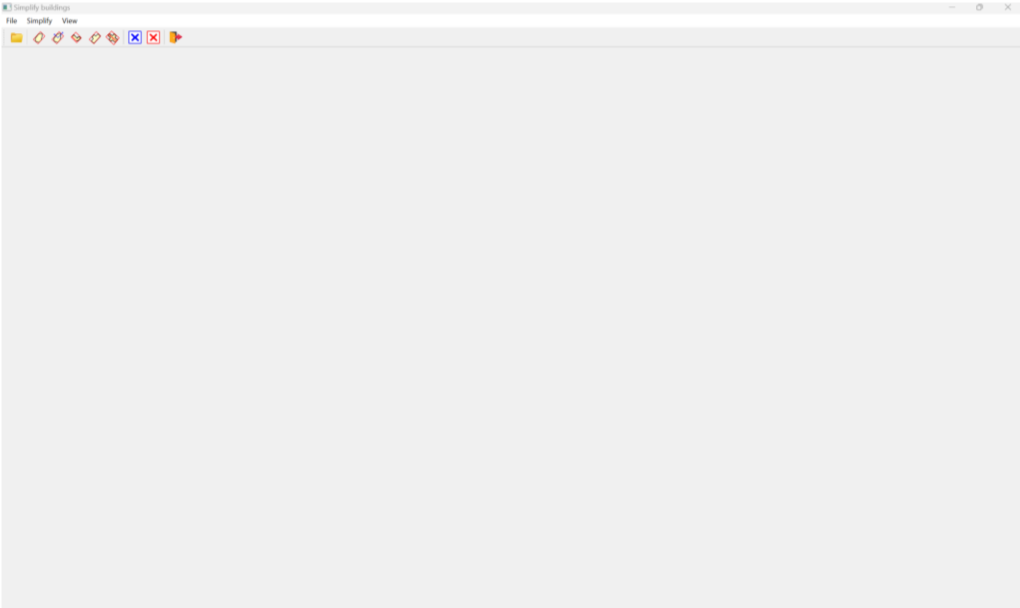
Za vstupní data jsme si vybrali různé části města Plzně. Tyto oblasti byly volené tak, aby splňovali podmínky úlohy. Data budov byla stažena z ArcGIS Online a poté byly zvolené části vybrány a sw. ArcGIS Pro a vyexportovány do formátu shapefile.

4. VÝSTUPNÍ DATA:

Výstupem tohoto úkolu je aplikace, která generalizuje tvar budov využitím různých algoritmů. U každé metody je zjištěno hodnotící kritérium, které se vypisuje do terminálu.

5. VYTVOŘENÁ APLIKACE:

Grafické rozhraní bylo vytvořeno s využitím frameworku QT, kde bylo vytvořeno grafické rozložení (umístění tlačítek, ...). Toto rozhraní bylo zkonvertováno do kódu programovacího jazyka Python.



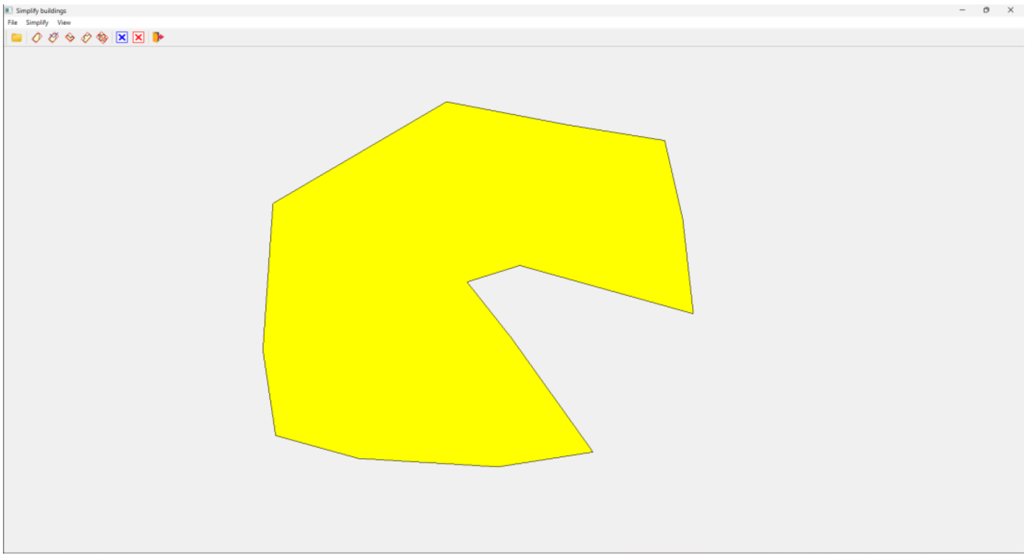
Obrázek 1: Grafické rozhraní

V grafickém rozhraní bylo vytvořeno pět tlačítek:

- : otevření souboru (umožňuje pouze otevřít soubory formátu *shapefile*),
- : spuštění metody *Minimum Bounding Rectangle,*
- : spuštění metody *PCA,*
- : spuštění metody *Longest Edge,*
- : spuštění metody *Wall Average,*
- : spuštění metody *Weighted Bisector,*
- : odstranění výsledků,
- : odstranění všech prvků,
- : vypnutí aplikace.

  Všechny tyto možnosti jsou také spustitelné z MenuBar.

V grafickém rozhraní si uživatel může sám vytvořit polygon nebo si jej může naimportovat z formátu shapefile.

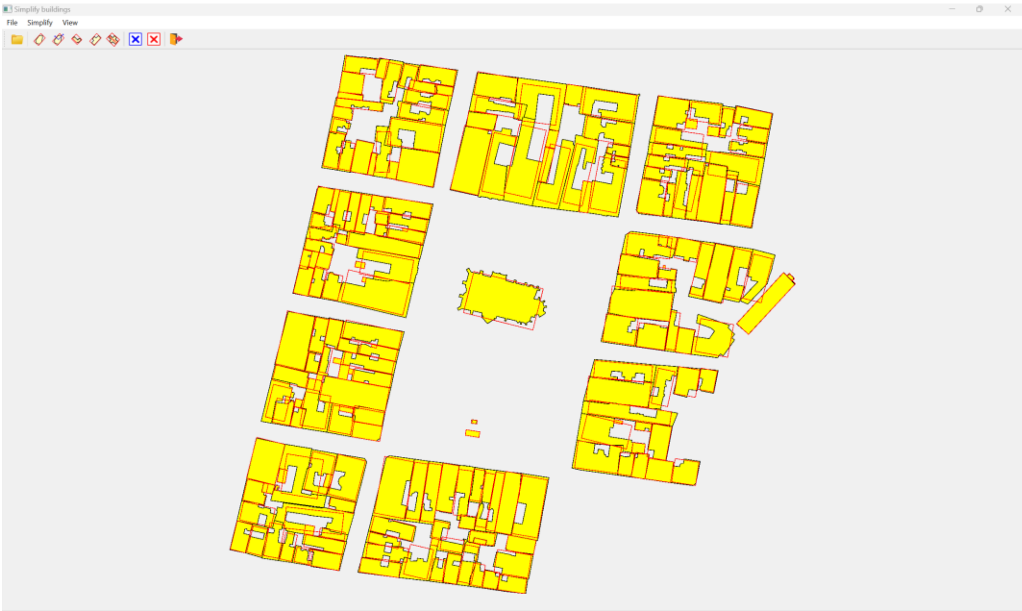


Obrázek 2: Grafické rozhraní – polygon nakreslený uživatelem

Obrázek 3: Grafické rozhraní – import polygonů

Po zvolení určité metody se uživatelovi zobrazí generalizovaný tvary budov:

1. Minimum bounding rectangle – červené obrysy



Obrázek 4: Grafické rozhraní – minimum bounding rectangle

2. PCA – modré obrysy

Obrázek 5: Grafické rozhraní – PCA

3. Longest Edge – šedé obrysy

Obrázek 6: Grafické rozhraní – Longest edge

4. Wall Average – zelené obrysy

Obrázek 7: Grafické rozhraní – Wall average

5. Weighted Bisector – tmavě zelené obrysy

Obrázek 8: Grafické rozhraní - Weighted bisector

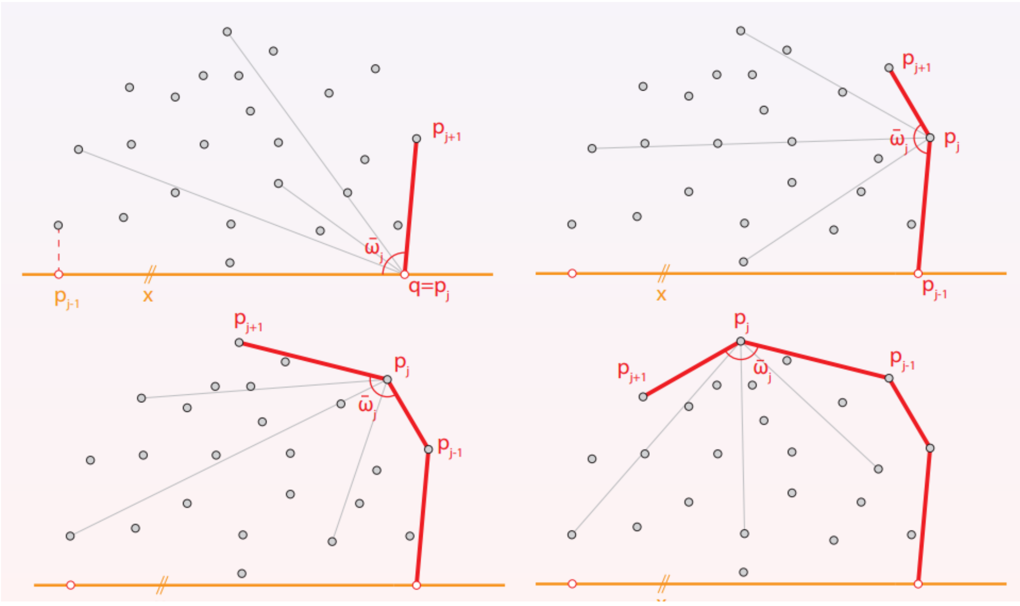
6. Postup zpracování:

Hlavním cílem tohoto úkolu bylo vytvořit algoritmy generalizující tvary budov na obdélníky.

a. *Vytvoření konvexní obálky metodou Jarvis Scan*:

Tato metoda připomíná postup balení dárku do papíru. Předpokladem tohoto algoritmu je, že množina bodů *P* neobsahuje tři kolineární body.

Tento algoritmus vezme poslední dva body a k němu se hledá třetí bod tak, aby maximalizoval úhel. Tento bod je poté přidán do obálky.



Obrázek 9: Metoda konstrukce konvexní obálky Jarvis Scan

Nalezení pivota q,q = min(y_i)

Přidej q → konvexní obálky

Inicializuj: $p_{j-1} \in X$, $p_j = q$, $p_{j+1} = p_{j-1}$

Opakuj, dokud $p_{j+1} \neq q$:

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

  Přidej p_{j+1} → přidej do konvexní obálky

  $p_{j-1} = p_j$; $p_j = p_{j+1}$

b. *Vytvoření min-max boxu*:

Min-max box je obdélník tvořený na základě minimálních a maximálních souřadnic bodů prvku. Tyto minimální/maximální souřadnice poté vytvoří hrany obdélníka.

Nalezni x_{min}, x_{max}, y_{min}, y_{max}

Vytvoř vrcholy: $v_1 = [x_{min}, y_{min}]$

$v_2 = [x_{min}, y_{max}]$

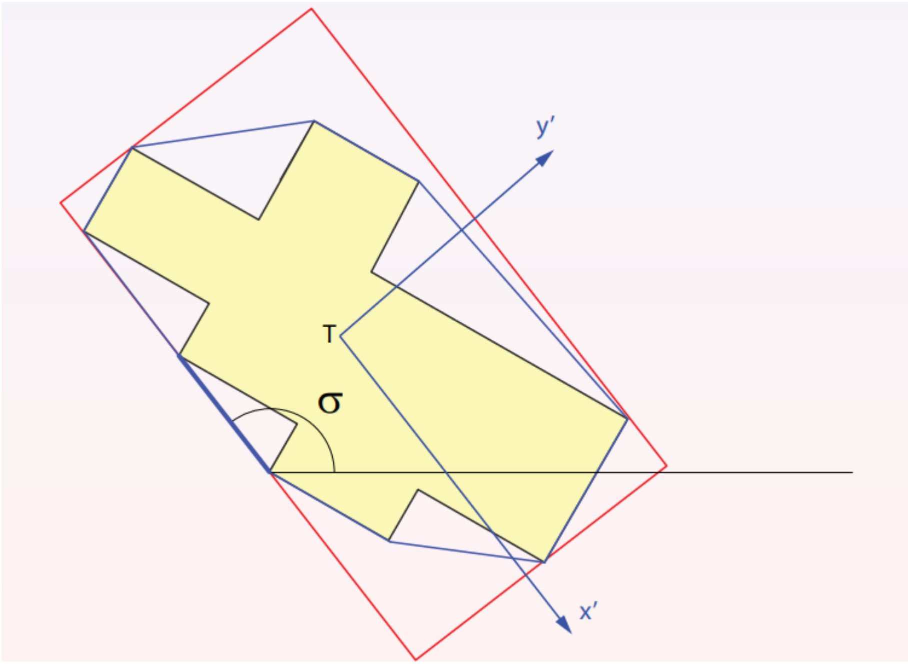
$v_3 = [x_{max}, y_{max}]$

$v_4 = [x_{max}, y_{max}]$

Z nadefinovaných vrcholů vytvoř obdélník

c. ***Metoda Minimum bounding rectangle:***

Tato metoda tvoří obdélník s minimální plochou. Platí zde, že aspoň jedna strana vytvořeného obdélníka je kolineární s konvexní obálkou vstupních bodů.



Obrázek 10: Metoda Minimum bounding rectangle

Najdi konvexní obálku
Inicializuj *R* = MMB(P), *A_* = A(MMB(P))
Opakuj pro každou hranu *e* konvexní obálky:
    Spočti směrnic *σ* hrany *e*
    Otoč *P* o -*σ* : $P_0 = R(-\sigma)P$
    Najdi MMB(P_0) a urči A(MMB(P_0))
    Pokud A < A_
        A_=A, MMB_=MMB(P_0), σ_=σ
R = R(σ_)MMB_

d. *PCA*

Tato metoda používá k nalezení hlavních směrů singulární rozklad.

→ Kovarianční matice

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

→ Singulární rozklad

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

→ Matice U,V: vlastní vektory CC^T a C^TC, jednotkové (cos rotací σ)

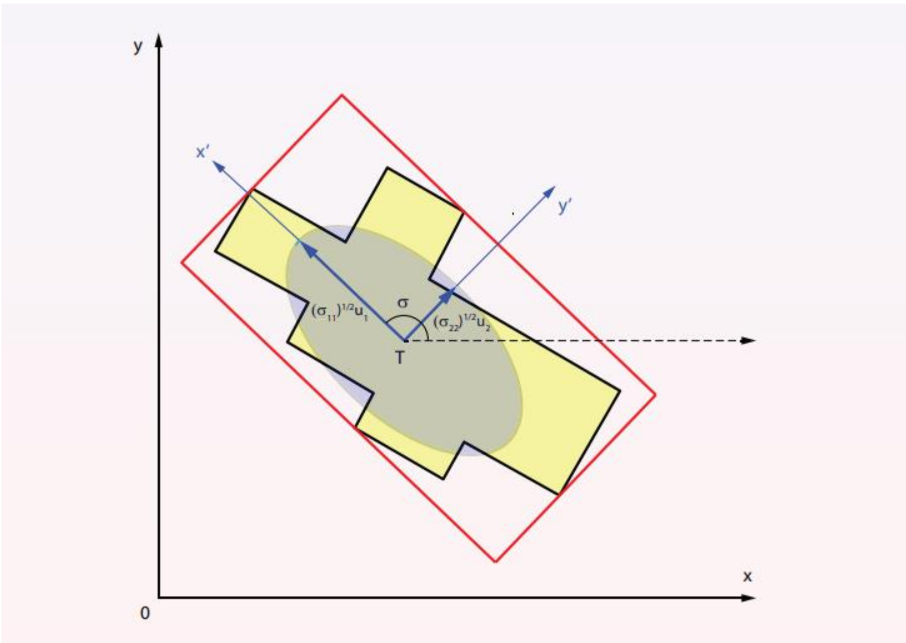
$$U = V = \begin{bmatrix} \cos\sigma & -\sin\sigma \\ \sin\sigma & \cos\sigma \end{bmatrix}.$$

→ Matice Σ: singulární hodnoty, čtverce vlastnícg 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 ±ω

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



Obrázek 11: Metoda Winding Number Algorithm

Inicializuj x a y a naplň je náležitými souřadnicemi všech bodů vstupního polygonu
Z těchto hodnot vytvoř pole P
Vypočti kovarianční matici C pole P
Udělej singulární rozklad matice C: [U, S, V] = svd(C)
Vypočti úhel rotace: σ = arctan(V_{21}, V_{11})
Vytvoř MMB

e. *Metoda Wall Average*

U této metody se na každou stranu aplikuje operace *mod(π/2)*. Ze zbytků hodnot je spočten vážený průměr přičemž je jeho váhou délka strany. Nejprve je určena směrnice σ_i všech stran a poté spočteny vnitřní úhly ω_i.

→ Pro každý vrchol p_i spočteme vnitřní úhly

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

→ Výpočet násobku *π/2*

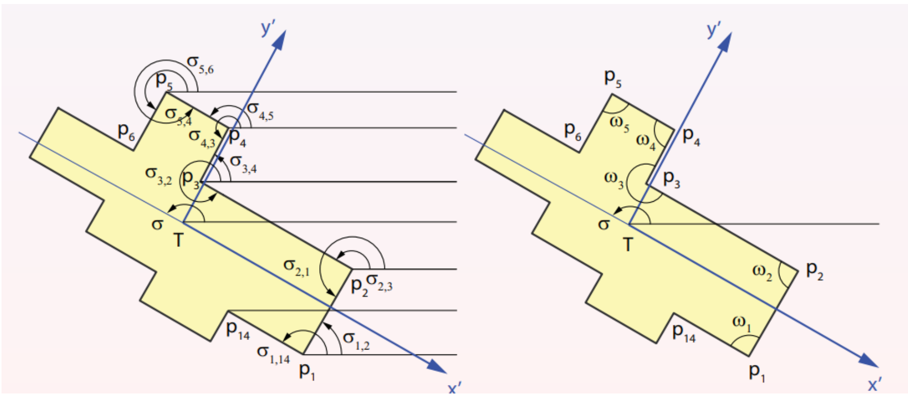
$$k_i = \frac{2\omega_i}{\pi}.$$

→ Orientovaný zbytek po dělení

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

→ Hlavní směr budovy

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

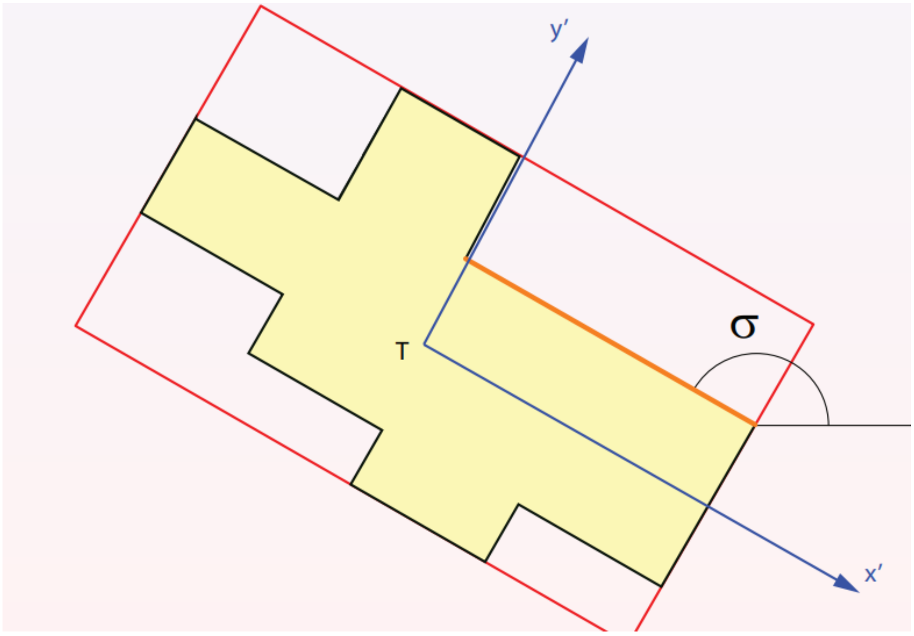


Obrázek 12: Metoda Wall Average

Vypočti počáteční směr
Vypočti směry pro všechny segmenty prvku
  Vypočti směr hrany prvku
  Vypočti délku hrany
  Vypočti rozdíl mezi touto hranou a hranou předchozí
  Vypočti orientovaný zbytek
Vypočti σ pomocí váženého průměru
Vytvoř MMB

f. Longest Edge

Metoda Longest Edge volí první hlavní směr budovy ve směru její nejdelší hrany a druhý směr kolmý na ni.



Obrázek 13: Metoda Longest Edge

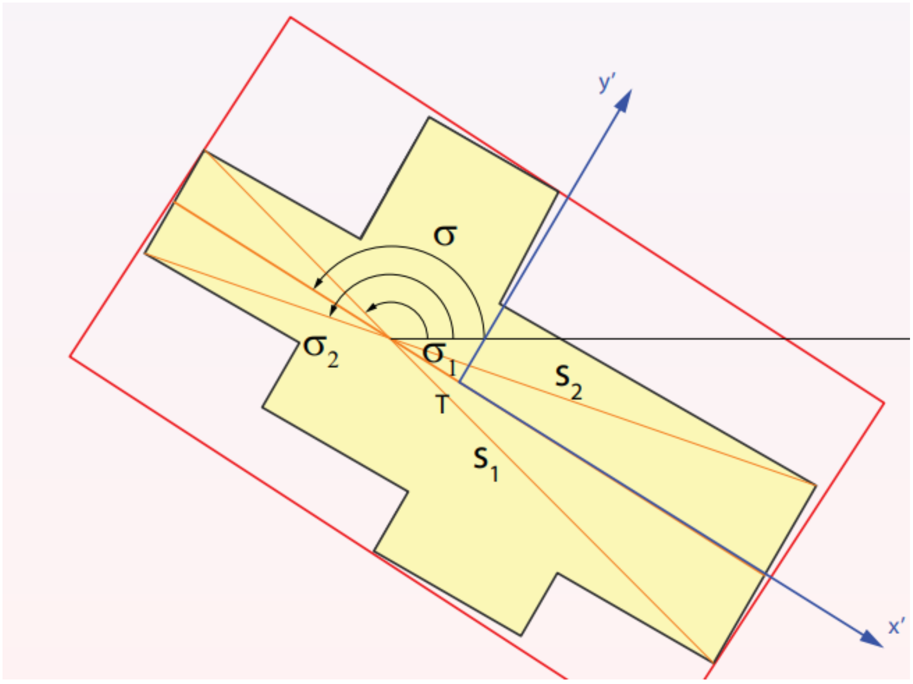
Zjisti nejdelší hranu budovy
Vypočti její směr
Vytvoř MMB

g. *Weighted Bisector*

V této metodě se hledají dvě nejdelší úhlopříčky, směrnice σ_1, σ_2 a délky s_1, s_2. Hlavní směr je poté dán váženým průměrem těchto hodnot.

→ Vážený průměr

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



Obrázek 14: Metoda Weighted Bisector

Inicializuj u_{1MAX}, u_{2MAX}

Zjisti vzdálenost mezi všemi nesousedními uzly segmentu

  Pokud je vzdálenost větší než u_{1MAX}

    Ulož tuto hodnotu do proměnné u_{1MAX}

    Ulož si souřadnicové rozdíly bodů této spojnice

  Nebo pokud je vzdálenost větší než u_{2MAX}

    Ulož tuto hodnotu do proměnné u_{2MAX}

    Ulož si souřadnicové rozdíly bodů této spojnice

Vypočti směry obou úhlopříček a následně výsledný směr hlavní osy pomocí váženého průměru

Vytvoř MMB

7. VÝSLEDKY:

a. *Historické centrum města*

b. Intravilán – sídliště

c. *Intravilán – izolovaná zástavba*

d. *Výsledné hodnotící kritérium*

METODA	Historické centrum města	Intravilán sídliště	Intravilán izolovaná zástavba
MAER	3.97315°	2.35181°	0.97764°
PCA	7.70020°	7.45282°	6.20701°
Wall Average	4.31694°	2.7318°	1.23019°
Longest Edge	4.05866°	2.42742°	1.01174°
Weighted bisector	8.88656°	5.45463°	7.01255°

8. ZÁVĚR:

Bylo vytvořené grafické rozhraní, kde si uživatel může sám nakreslit polygon nebo si polygony naimportovat ze souboru shapefile. Do tohoto rozhraní byly zaimplementovány metody Minimum Bounding Rectangle, PCA, Wall Average, Longest Edge a Weighted Bisector.

9. ZDROJE:

Obrázky [6], [7], [8], [9], [10], [11] a informace o průběhu výpočtu:

Konvexní obálka množiny bodů. Online. BAYER, Tomáš. Konvexní obálka množiny bodů. 2024. Dostupné z: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4_new.pdf. [cit. 2024-05-01].

V Plzni dne 9.5.2024 Kateřina Chromá, Štěpán Šedivý