Technická zpráva úkol č. 1: Point Position Problem

Miroslav Hrubeš, Lucie Peterková

Praha 2023

1 Zadání

Uloha č. 2: Generalizace budov
Vstup: množina budov $B = \{Bi\}$*, budova* $Bi = \{P_{i,j}\}$
Výstup: $G(Bi)$*.*

Ze souboru načtěte vstupní data představová lomovými body budov. Pro tyto účely použijte vhodnou datovou sadu, např. ZABAGED.

Pro každou budovu určete její hlavní směry metodami:

- Minimum Area Enclosing Rectangle
- Wall Average

U první metody použijte některý z algoritmů pro konstrukci konvexní obálky. Budovu nahraďte obdélníkem se středem v jejím těžišti orientovaným v obou hlavních směrech, jeho plocha bude stejná jako plocha budovy. Výsledky generalizace vhodně vizualizujte.

Odhadněte efektivitu obou metod, vzájemně je porovnejte a zhodnoťte. Pokuste se identifikovat, pro které tvary budov dávají metody nevhodné výsledky, a pro které naopak poskytují vhodnou aproximaci.

2 Bonusové úlohy

Nebyly řešeny žádné bonusové úlohy.

3 Generalizace budov

Kartografická generalizace, nebo též zjednodušení, je důležitým úkonem, který se používá v celém procesu tvorby mapy, od sběru dat v terénu až po konečnou podobu mapy. Důvodem pro použití kartografické generalizace je měřítko, protože mapa představuje zmenšený obraz reality. Aby byla mapa srozumitelná, čitelná a zároveň geograficky věrná a geometricky přesná, je nutné některé prvky zjednodušit. Použité postupy, metody a parametry závisí vždy na měřítku mapy - čím menší měřítko, tím více musí být mapa zjednodušena. Dále na účelu mapy - různé typy map vyžadují různou úroveň zjednodušení jednotlivých prvků a v neposlední řadě na cílových uživatelích (Miklín 2018). Obecně se tedy dá generalizace označit jako proces snižování množství dat a přizpůsobení množství informací danému měřítku a tématu (Muller 1991).

Generalizace budov je složitá a zahrnuje mnoho aspektů. Jedním z hlavních problémů je způsob reprezentace budov na mapě, protože budovy mají různé tvary a velikosti a navíc se mohou nacházet v různých prostorových vztazích. Vzhledem k různým měřítkům map se také liší úroveň detailů a zobrazovaných prvků na mapě. Pro velké měřítko je obvykle nutné zobrazovat jednotlivé budovy podrobněji, zatímco pro menší měřítko se většinou používají zjednodušené formy a symboly.

4 Konvexní obálka

Konvexní obálka (convex hull) je základním geometrickým pojmem, který označuje nejmenší konvexní oblast, která zahrnuje danou množinu bodů v rovině nebo v prostoru. V kontextu počítačové kartografie se konvexní obálka používá k generalizaci (zjednodušení) složitých geografických prvků, jako jsou například budovy. V této metodě se nejprve vytvoří konvexní obálka pro každou budovu na základě jejích původních polygonálních hranic. Poté se vytvoří nové, zjednodušené polygonální hrany tak, že se spojí body konvexní obálky. Tento proces může být opakován pro různé úrovně zjednodušení, čímž se vytvoří hierarchická struktura zjednodušených reprezentací budov. Výhodou použití konvexní obálky pro generalizaci budov je, že výsledná reprezentace je konvexní tudíž neobsahuje ostré výčnělky. To znamená, že lze jednoduše určit, zda se dvě budovy překrývají nebo ne. Nevýhodou může být, že se může vytvářet příliš jednoduchá reprezentace budov, která nezachycuje dostatečně jejich původní tvar a detaily.

Existuje několik algoritmů pro tvorbu konvexní obálky, některé z nejznámějších jsou: *Graham scan, Quickhull, Jarvis scan, Divide and Conquer*

V této práci byl použit algoritmus *Jarvis scan.*

4.1 Jarvis scan

Algoritmus *Jarvis scan*, také známý jako Gift wrapping algorithm, je základní algoritmus pro výpočet konvexní obálky množiny bodů v rovině, který byl představen již v roce 1973. Jeho princip spočívá v tom, že postupně hledá nejvzdálenější bod od aktuálního bodu a spojuje ho s konvexní obálkou. Obecný postup by se dal popsat následovně:

Nejdříve se vybere počáteční bod. Tento bod se vloží do konvexní obálky a označí se jako aktuální bod. Poté se k němu nalezne bod s největším úhlem a přidá ho do konvexní obálky. Tento bod se nyní stává aktuálním bodem a postup se opakuje, dokud se nevrátíme na začátek (Jarvis 1973).

Na vstupu pro *Jarvis Scan* je dána množina bodů P_i. Poslední body tvořené konvexní obálkou H

se označují jako P_{j-1} a P_j. Bod P_{j+1} značí poslední přidaný bod do H. Musí platit maximalizace úhlu $\omega(P_{j-1}, P_j, P_i)$, přičemž daný bod $P_i \notin H$:

$$P_{j+1} = \max : \omega(P_{j-1}, P_j, P_i)$$

Úhly $\omega(P_{j-1}, P_j, P_i)$ se spočítají pomocí vzorce pro výpočet odchylky dvou vektorů $\vec{u}$ a $\vec{v}$, přičemž $\vec{u}$ je směrový vektor přímky $P_{j-1}P_j$ a $\vec{v}$ směrový vektor přímky P_jP_i:

$$\cos\omega = \frac{\vec{u}\vec{v}}{\|\vec{u}\|\|\vec{v}\|}$$

Algoritmus *Jarvis scan* má složitost nlogh, kde n je počet bodů a h je počet bodů v konvexní obálce. Je tedy vhodný pro malé množiny bodů, ale pro velké množiny bodů může být pomalý.

Psedukódem se dá tento algoritmus zapsat následovně:

Algorithm 1 *Jarvis Scan*

1: Nalezení bodu P_{min} s minimální y souřadnicí: $P_{min} = \min(y_i)$.
2: Přidání bodu P_{min} do tvořené konvexní obálky H: $P_{min} \in H$.
3: Inicializace posledních dvou bodů H: $P_{j-1} = [-\infty, \min(y_i)]$, $P_j = P_{min}$.
4: Inicializace přidávaného bodu do H: $P_{j+1} = P_{j-1}$.
5: **Dokud** $P_{j+1} \neq P_{min}$:
6: **Pro** každý vstupní bod P_i:
7: Spočítání úhlu $\omega(P_{j-1}, P_j, P_i)$.
8: Nastavení $P_{j+1} = \max \omega(P_{j-1}, P_j, P_i)$.
9: Přidání P_{j+1} do konvexní obálky H: $P_{j+1} \in H$.
10: Aktualizace posledních dvou bodů H: $P_{j-1} = P_j$, $P_j = P_i$.

5 Minimum Area Enclosing Rectangle

Metoda *Minimum Area Enclosing Rectangle* (MAER) obecně hledá nejmenší možný obdélník, který obaluje celou množinu bodů (jejich konvexní obálku), neboli všechny body množiny P_i budou na hraně či uvnitř obdélníku jehož strany jsou rovnoběžné s osami souřadnic.

Po vytvoření konvexní obálky H přistoupíme k tvorbě *Minimum Bounding Box* - obdélníku, který obklopuje všechny lomové body budovy a jeho plocha je minimální. Tuto tvorbu provádíme opakovanou rotací množiny vstupních lomových bodů P_i o úhel $-\sigma$. Tento úhel je dán zápornou hodnotou směrnice σ určité hrany konvexní obálky H. Rotace má za následek "postavení"konvexní obálky na danou hranu.

Hrana konvexní obálky H je tvořena body $C_i = [x_i, y_i]$ a $C_{i+1} = [x_{i+1}, y_{i+1}]$. Směrnici hrany spočítáme následovně:

$$\tan \sigma = \frac{\Delta y}{\Delta x}$$

Poté vstupní lomové body a konvexní obálku rotujeme o $-\sigma$ pomocí matice rotace:

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} cos(-\sigma) & sin(-\sigma) \\ -sin(-\sigma) & cos(-\sigma) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

V této poloze se konstruuje *Minimum Bounding Box* konvexní obálky. V první řadě je nutné najít vrcholy tohoto obdélníku, které jsou tvořeny pomocí horních a dolních extrémů souřadnic x_i a y_i. Pro vytvořený *Minimum Bounding Box* se spočítá plocha S. Takto vytvořený *Minimum Bounding Box* následně otočíme zpět do původní polohy pomocí výše uvedené matice rotace. Uhel rotace je σ, a platí, že otočením se plocha S nezmění.

Výsledný *Minimum Area Enclosing Rectangle* po iteraci přes každou hranu konvexní obálky je ten, který má nejmenší plochu S.

Psedukódem se dá tento algoritmus zapsat následovně:

Algorithm 2 *Minimum Area Enclosing Rectangle*

1: Vytvoření konvexní obálku H.
2: Vytvoření počáteční aproximace *Minimum Bounding Box* MBB_{min} na základě konvexní obálky H a spočítání jeho plochy S_{min}.
3: Inicializace $\sigma_{min} = 0$.
4: **Pro** každou hranu konvexní obálky H se opakuje:
5: Spočítání směrnice σ hrany konvexní obálky H.
6: Otočení konvexní obálky H o úhel -σ.
7: V této poloze vytvoření *Minimum Bounding Box* MBB a spočítání jeho plochy S.
8: **Pokud** $S < S_{min}$:
9: $S_{min} = S$, $MBB_{min} = MBB$, $\sigma_{min} = \sigma$.
10: Otočení MBB_{min} zpět do původní polohy o úhel σ.
11: Spočítání plochy polygonu S_b.
12: Upravení MBB_{min} tak, aby: $S(MBB_{min}) = S_b$.

6 Wall Average

Metoda *Wall Average* byla představena v roce 1998 a je další technika pro generalizaci budov. Metoda obecně spočívá v tom, že je nejprve vytvořena aproximace budovy pomocí konvexní obálky nebo *Minimum Bounding Box* . Poté se provede průměrování délek a směrů jednotlivých stěn v rámci budovy. Tím se získá průměrná délka a směr stěn. Nakonec se budova upraví tak, aby každá ze stěn měla průměrnou délku a směr. Výsledkem je zjednodušená verze budovy, která je stále dostatečně podobná původní budově.

Hlavním principem je vypočítání ukazatele orientace hran budovy. Pro měření této orientace se používá obecný princip, který spočívá v tom, že se vezme v úvahu orientace každé hrany $mod(\frac{\pi}{2})$ (tj. mezi 0 a $\frac{\pi}{2}$) a následně se vypočítá průměr těchto směrů. Váhy pro každou hranu jsou určeny délkou této hrany (Duchene 2003).

V prvním kroku algoritmu se pro libovolnou hranu polygonu představující budovu vypočítá směrnice σ'. Následně se spočítá směrnice σ_i pro každou hranu budovy a hodnoty σ_i se sníží o hodnotu σ':

$$\Delta\sigma_i = \sigma_i - \sigma'$$

Ze spočítaných hodnot $\Delta\sigma_i$ se vypočítá dolů zaokrouhlený podíl k_i:

$$k_i = \frac{2\Delta\sigma_i}{\pi}$$

V dalším kroku se již přistupuje k samotnému výpočtu zbytku r_i, jehož hodnota určuje, zda je hrana odchýlená spíše od vodorovného směru ($r_i < \frac{\pi}{4}$) nebo spíše od vertikálního směru ($r_i > \frac{\pi}{4}$):

$$r_i = \Delta\sigma_i - k_i\frac{\pi}{2}$$

Jak bylo zmíněno, samotná orientace budovy σ je určena pomocí váženého průměru zbytků r_i, kde váhou je délka hran, označená jako d_i, a kde d značí součet všech délek hran:

$$\sigma = \sigma' + \frac{\sum_{i=1}^{n} r_i d_i}{d}$$

Podobně jako u předchozího algoritmu se budova otočí o úhel -σ, v této poloze se konstruuje *Minimum Bounding Box*, ten se otočí zpět o úhel σ a upraví se tak, aby jeho plocha S byla totožná s plochou budovy S_b.

Psedukódem se dá tento algoritmus zapsat následovně:

Algorithm 3 Wall Average

1: Vypočítání směrnice σ' pro libovolnou hranu budovy:
2: **Pro** každou hranu budovy:
3: Vypočítání směrnice σ_i.
4: Vypočítání $\triangle\sigma'$
5: Vypočítání k_i
6: Vypočítání r_i
7: Vypočítání výsledné směrnice σ
8: Otočení všech hran budovy o úhel $-\sigma$
9: Vytvoření MBB
10: Otočení MBB o úhel σ
11: Úprava velikosti obsahu MBB na $S_{MMB} = S_b$

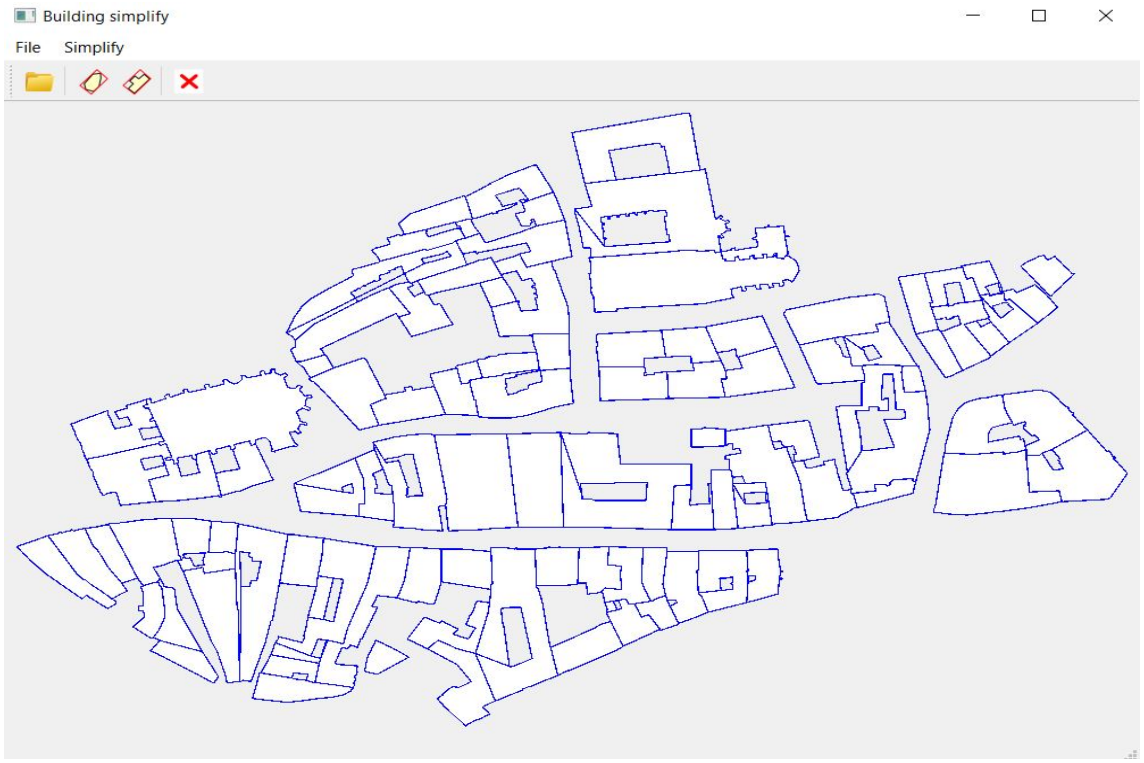
7 Vstupní a výstupní data

Vstupem aplikace je soubor obsahující polygony budov ve formátu shapefile *(.shp)*. Funkčnost aplikace byla testována na 3 datasetech budov, kde každý obsahuje přibližně 100 prvků. Přiložené datasety obsahují různé typy zástavby pro ilustraci fungování algoritmů. Dataset *prahacentrum.shp* obsahuje budovy, které jsou součástí historického centra prahy. Dataset *prahaintravilan.shp* obsahuje budovy panelákového typu a dataset *praharoddomy.shp* obsahuje jednotlivé rodinné domy

Aplikace nenabízí výstupní data v podobě souboru k uložení. Výsledkem je grafické zobrazení polygonů generalizovaných budov dle vybrané metody a původní dataset pro porovnání úspěšnosti generalizace.

8 Aplikace

Obrázek 1 ukazuje grafické rozhraní vytvořené aplikace pro generalizaci budov. Hlavní okno zobrazuje nahranou polygonovou vrstvu budov ve tvaru *.shp*. V horní části nachází záložka pro nahrání *.shp* souboru z uložiště počítače. Dále se zde nacházejí tlačítka pro zvolení metody generalizace budov. Prvním je metoda *Minimum Area Enclosing Rectangle* a druhá je metoda *Wall average*. Jako polsední se na záložce nachází tlačítko Clear, které vymže výsledky generalizace. v záložce File se nachází tlačítko Exit, které ukončí celou aplikaci.



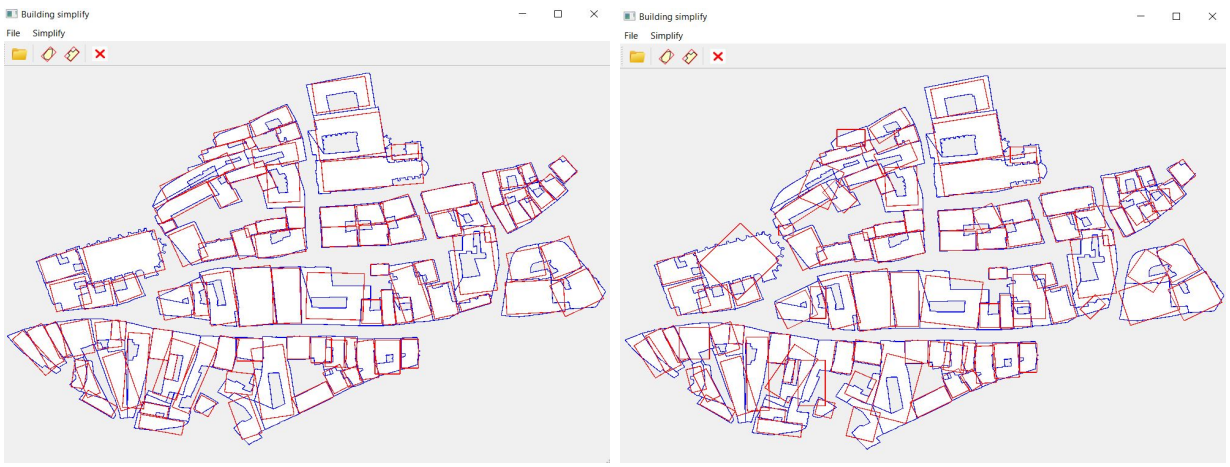
Obrázek 1: Building simplify, zdroj: autoři.

9 Výsledky

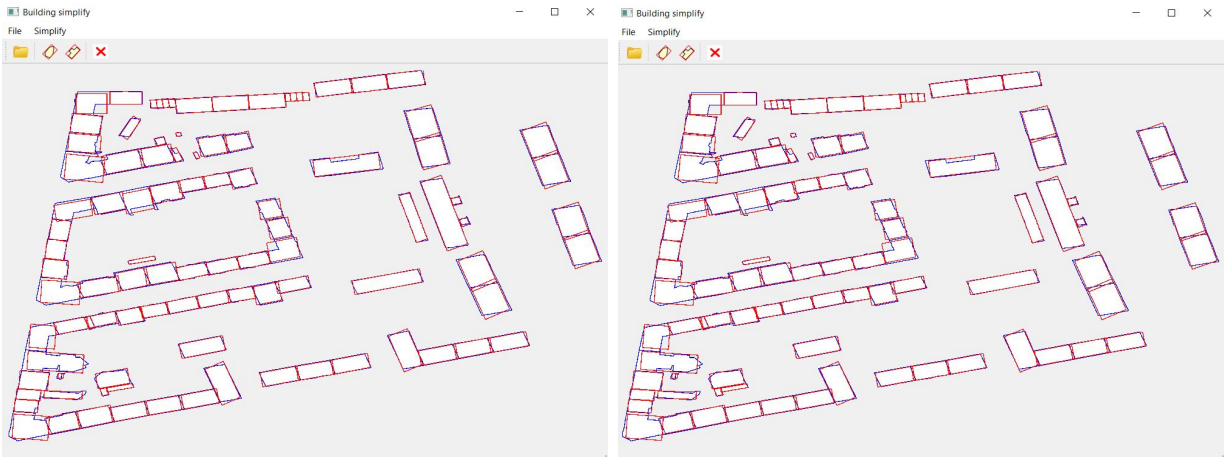
Obrázky 2, 3 a 4 představují výsledek a porovnání generalizace budov v rozdílných datasetech při použití metod *Minimum Area Enclosing Rectangle* a *Wall average*.

Dle subjektivního vizuálního hodnocení lze obecně výsledky hodnotit jako velmi dobré. Obě metody nejlépe generalizují budovy z datasetu Intravilán, což je způsobeno převládajícím jednoduchým a pravidelným tvarem budov. Horší výsledek je patrný u datasetu Rodinné domy a to především u metody *Wall average*, kde jsou patrné výchylky u budov nepravidelného nebo specifického (L, E, atd.) tvaru budov. U metody *Minimum Area Enclosing Rectangle* není pozorován žádný výrazný nedostatek.

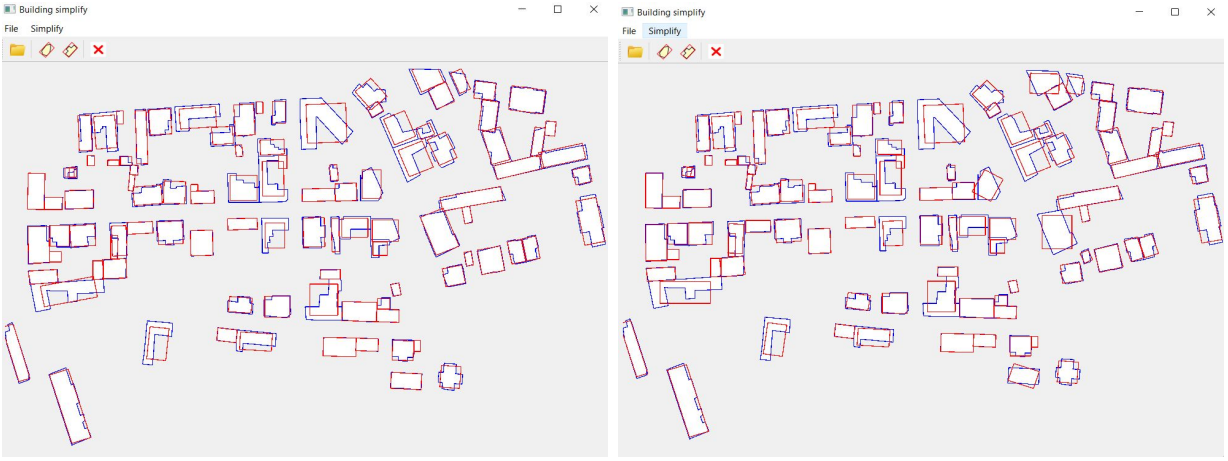
Nejméně přesné výsledky jsou viditelné u datasetu Historické centrum a to především kvůli rozmanitým tvarům budov s nemalým množstvím různých výběžku. Výrazně horších výsledků zde dosahuje taktéž metoda *Wall average*, která si s velkým množstvím budov především protáhlého a sbíhavého tvaru nedokázala správně poradit. Metoda *Minimum Area Enclosing Rectangle* dosahuje viditelně lepší výsledky u tohoto datasetu, nicméně i zde je možné nalézt několik nesprávně generalizovaných budov. U tohoto datasetu je také horší možnost provedení hodnocení a to kvůli poloze budov, které jsou velmi blízko u sebe, čímž dochází u vytvořené generalizace k velkému množství překryvů.



Obrázek 2:
Dataset: Historické centrum. Vlevo: *Minimum Area Enclosing Rectangle*, vpravo: *Wall average*
zdroj: autoři.



Obrázek 3:
Dataset: Intravilán. Vlevo: *Minimum Area Enclosing Rectangle*, vpravo: *Wall average* zdroj: autoři.



Obrázek 4:
Dataset: Rodinné domy. Vlevo: *Minimum Area Enclosing Rectangle*, vpravo: *Wall average* zdroj: autoři.

10 Dokumentace

Aplikace je tvořena třemi skripty. Uživatelské rozhraní je tvořeno skriptem *mainform.py*, jehož základ byl vytvořen pomocí softwaru *QT Creator* . Vizualizaci a vykreslování objektů zajišťuje skript *draw.py*. Skritp *algorithms.py* definuje matematické metody pro správné provedení zvolených analýz.

Metody skriptu *mainform.py* :

- **setupUi**: inicializuje a nastavuje různé prvky uživatelského rozhraní, jako např. velikost okna, hlavní widget, menu, lištu nástrojů a akce (např. otevření souboru, ukončení aplikace, vykreslení polygonu, analýza pozice bodů a polygonů apod.).
- **retranslateUi**: nastavení textů pro různé prvky uživatelského rozhraní.

Metody skriptu *draw.py* :

- **setPath**: načte cestu ke souboru .shp pomocí dialogového okna, přečte soubor a vytvoří polygon.
- **mousePressEvent**: spustí se po kliknutí levým tlačítkem myši. Metoda uloží souřadnice bodu, do kterého se kliklo do proměnných x a y, přidá nový bod p do polygonu a překreslí obrazovku.
- **paintEvent**: spustí se při vykreslení okna. Vytvoří grafický objekt, nakreslí polygon a zvýrazní polygon obsahující daný bod.
- **switchsource**: přepíná mezi režimy přidání bodu a přidání vrcholu polygonu.
- **getPoint**: vrací aktuální pozici bodu jako QPointF.
- **getPolygon**: vrací standardní polygon jako seznam QPolygonF.
- **getResPol**: přidá výsledný polygon jako QPolygonF do seznamu.
- **clearResPol**: vymaže všechny výsledné polygony.

Metody skriptu *algorithms.py* :

- **getPointPolygonPositionR**: metoda pro zjištění pozice bodu q v polygonu pol. Vrací 1, pokud je bod uvnitř polygonu a 0, pokud je vně polygonu. Metoda využívá Ray-casting algoritmus.
- **get2LinesAngle**: metoda pro výpočet úhlu mezi dvěma přímkami reprezentovanými čtyřmi body. Vrací hodnotu v radiánech.
- **createCH**: metoda pro vytvoření konvexního obalu pomocí algoritmu Jarvis scan. Vrací QPolygonF objekt obsahující vrcholy konvexního obalu.
- **rotate**: metoda pro rotaci polygonu pol o úhel sig. Vrací QPolygonF objekt s novými souřadnicemi vrcholů.
- **minMaxBox**: metoda pro vytvoření min-max boxu. Vrací QPolygonF objekt obsahující vrcholy min-max boxu a jeho plochu.
- **computeArea**: metoda pro výpočet plochy polygonu pol. Vrací hodnotu plochy polygonu.
- **resizeRectangle**: metoda pro změnu velikosti obdélníku er tak, aby se přizpůsobil polygonu pol. Metoda vrací QPolygonF objekt s novými souřadnicemi vrcholů obdélníku.

11 Závěr

Výsledkem je funkční aplikace implementující metody *Minimum Area Enclosing Rectangle* a *Wall average* pro generalizaci budov. Dané metody využívají funkce *Jarvis scan* pro výpočet konvexní obálky, která je nedílným prvkem správného provedení generalizace.

Práce taktéž přináší porovnání obou použitých metod, nicméně pro přesné stanovení přesnosti by bylo nutné použít datasety s větším počtem budov převážně různorodých tvarů. Zároveň by bylo vhodné použít exaktní metodu, pro přesnější stanovení přesností výsledků.

Zdokonalení aplikace je možné především s využitím dalších metod generalizace a metod pro tvorbu konvexní obálky, což by poskytlo další metodu, která by mohla být porovnávána a diskutována.

Problém představují polygony, již jsou tvořeny vícero částmi (multipart polygony) jako například "kruhové"budovy s blokem uvnitř. Aplikace tato data nedokáže správně přečíst, čímž vzniká mezi určitými body úsečka rozdělující polygony, což může způsobit nepřesné provedení generalizace.

12 Zdroje

Přednášky z předmětu *Algoritmy z počítačové kartografie.*

DUCHENE, C., BARD, S., BARILLOTt, X., RUAS, A., TREVISAN, J., HOLZAPFEL, F. (2003): Quantitative and qualitative description of building orientation. Institut Géographique National, COGIT Laboratory.

JARVIS, R. (1973):On the identification of the convex hull of a finite set of points in the plane, Information Processing Letters 2, 1, 18-21.

MIKLÍN, J., DUŠEK, R., KRTIČKA, L., KALÁB, O. (2018): Tvorba map. Ostrava: Ostravská univerzita.

MULLER, J. C. (1991). Generalization of Spatial Databases. In: D. J. MAGUIRE, D. J., GOODCHILD,M. F., RHINDS, D. W. Geographic Information Systems: Principles and Practice. 1, 457–475. Longman, London.