

Generalizace budov

Algoritmy digitální kartografie a GIS

Jakub Šnopl, Dennis Dvořák, Jakub Felenda

24. listopadu 2023

Obsah

1	Zadání	3
2	Bonusové úlohy	4
3	Použité algoritmy	4
3.1	Výpočet konvexní obálky	4
3.2	Generalizační algoritmy	4
3.2.1	Minimum Area Enclosing Rectangle	4
3.2.2	Wall Average	5
3.2.3	Longest edge	5
4	Vstupní data	6
4.1	Načtení WKT dat	7
5	Vzhled aplikace	7
6	Zhodnocení generalizačních metod	8
7	Dokumentace	10
7.1	Třída Algorithms	10
7.2	Třída Draw	11
7.3	Třída WKT	11

7.4	Třída MainForm : public QMainWindow	11
7.5	Třída sortPointsByX a třída sortPointsByY	11
8	Závěr	12
	Reference	13

1 Zadání

Úloha č. 2: Generalizace budov LOD0

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 představovaná lomovými body budov a proved'te generalizaci budov do úrovně detailu LOD0. Pro tyto účely použijte vhodnou datovou sadu, např. ZABAGED, testování proved'te nad třemi datovými sadami (historické centrum města, intravilán - sídliště, intravilán - izolovaná zástavba).

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 při generalizaci do úrovně LOD0 nahraďte obdélníkem orientovaným v obou hlavních směrech, se středem v jejím těžišti, jeho plocha bude stejná jako plocha budovy. Výsledky generalizace vhodně vizualizujte.

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

Hodnocení:

Krok	Hodnocení
Generalizace budov metodami Minimum Area Enclosing Rectangle a Wall Average	15b
Generalizace budov metodou Longest Edge.	+5b
Generalizace budov metodou PCA.	+8b
Generalizace budov metodou Weighted Bisector.	+10b
Implementace další metody konstrukce konvexní obálky.	+5b
Ošetření singulárního případu u při generování konvexní obálky.	+2b
Max celkem:	45b

Čas zpracování: 3 týdny.

2 Bonusové úlohy

- Generalizace budov metodou Longest Edge

3 Použité algoritmy

Vstupem je množina polygonů budov $B = \{B_i\}_{i=1}^n$, každá jednotlivá budova je množinou bodů $B_i = \{P_{i,j}\}_{j=1}^m$. Výstupem je pak generalizovaná množina polygonů budov $G(B_i)$ o úrovni generalizace LoD0.

3.1 Výpočet konvexní obálky

Konvexní obálka je takový mnohoúhelník H , který tvoří minimální ohraničení množiny bodů B_i , zároveň pokud spojíme libovolné dva body množiny bodů B_i , musí se celá úsečka nacházet uvnitř obálky. Konvexní obálka byla v této úloze hledána algoritmem *Jarvis Scan*.

Algoritmus 1: Jarvis Scan

```
Nalezení pivota  $q$ ,  $q = \min(y_i)$ 
Přidej:  $q \rightarrow H$ 
Inicializuj:  $p_{j-1} \in X$ ,  $p_j = q$ ,  $p_{j+1} = p_{j-1}$ 
do
    Nalezni:  $p_{j+1} = \arg \max_{p_i \in P} \angle(p_{j-1}, p_j, p_i)$ 
    Přidej:  $p_{j+1} \rightarrow H$ 
     $p_{j-1} = p_j$ ;  $p_j = p_{j+1}$ 
while  $p_{j+1} \neq q$ 
```

3.2 Generalizační algoritmy

Kartografická generalizace je takový proces, při kterém dochází ke zjednodušení a zveřejnění objektů a jejich vzájemných vztahů pro jejich grafické vyjádření v mapě. V této úloze byla generalizace provedena pomocí následujících algoritmů.

3.2.1 Minimum Area Enclosing Rectangle

Jedná se o algoritmus, který minimalizuje plochu obdélníku R opsaného konvexní obálce H generalizované budovy S .

Algoritmus 2: Minimum Area Enclosing Rectangle

Výpočet konvexní obálky $H = CH(S)$
Inicializuj $R = MMB(S)$, $\underline{A} = A(MMB)$
for každou hranu e obálky H **do**
 Spočti směrnici σ hrany e
 Otoč S o $-\sigma$: $S_r = R(-\sigma)S$
 Najdi $MMB(S_r)$ a urči $A(MMB(S_r))$
 if $A < \underline{A}$ **then**
 $\underline{A} = A$; $\underline{MMB} = MMB$; $\underline{\sigma} = \sigma$
 end
end
 $R = R(-\underline{\sigma})\underline{MMB}$

3.2.2 Wall Average

Pro každou hranu budovy S je vypočtena její směrnice na kterou je aplikována operace $\text{mod}(\frac{\pi}{2})$, následně je pomocí váženého průměru, kde váhou je délka strany, vypočten směr natočení $\bar{\sigma}$ generalizované budovy R .

Algoritmus 3: Wall Average

Výpočet σ první hrany budovy
for každou hranu Budovy S **do**
 Spočti směrnici σ_i hrany
 Redukuj $\sigma_{red} = \sigma_i - \sigma$
 Spočti $r_i = \sigma_{red} \% \frac{\pi}{2}$
end
Výpočet průměrné směrnice $\bar{\sigma} = \sigma + \sum_{i=1}^n \frac{r_i s_i}{s_i}$
 $S_r = R(-\sigma)S$
 $R_r = MMB(S_r)$
 $R = q * R(\sigma)R_r$

3.2.3 Longest edge

Nejdelší hrana polygonu budovy S udává hlavní směr generalizované budovy R , druhý hlavní směr je na první kolmý.

Algoritmus 4: Longest Edge

```
Inicializuj  $d_{max}, index$   
for každou hranu Budovy  $S$  do  
    Spočti délku hrany  $d$   
    if  $d > d_{max}$  then  
         $d_{max} = d; index = i$   
    end  
end  
Spočti  $\sigma_{index}$   $S_r = R(-\sigma_{index})S$   
 $R_r = MMB(S_r)$   
 $R = q * R(\sigma_{index})R_r$ 
```

4 Vstupní data

Pro testování generalizačních algoritmů byly využity celkem tři datové sady se třemi různými druhy městské zástavby - sídliště (sidli.csv), centrum (centr.csv) a vilovou čtvrtí (roddomy.csv). Všechna použitá data jsou k nalezení v Github repozitáři projektu.

Vstupní polygonová data jsou ve formátu WKT [2]. Zdrojem je datová sada RÚIAN (*Registr územní identifikace, adres a nemovitostí*). Stáhnutí dat proběhlo pomocí pluginu RÚIAN v softwaru QGIS. Ze stažených dat byla využita vrstva *stavební objekty*.



Obrázek 1: Ukázka vstupních dat, rotováno o 90°

Formát dat pro jeden polygon je následující:

"MULTIPOLYGON((($-\mathbf{Y}_1$ $-\mathbf{X}_1$, $-\mathbf{Y}_2$ $-\mathbf{X}_2$, ... , $-\mathbf{Y}_n$ $-\mathbf{X}_n$, $-\mathbf{Y}_1$ $-\mathbf{X}_1$)))"

,kde n je počet vrcholů daného polygonu; \mathbf{Y} jsou souřadnice Y v souřadnicovém systému JTSK; \mathbf{X} jsou souřadnice X v souřadnicovém systému JTSK (EPSG:5514).

4.1 Načtení WKT dat

O načtení souboru *csv* se stará třída *WKT*. Jejím úkolem je načíst *csv* data ze souborů. Třída ukládá výsledek do vektorů polygonů a taktéž ukládá maximální i minimální x a y souřadnici z načteného souboru. Postup je následující:

- Načtení souboru *csv*.
- Cyklus přes všechny řádky souboru.
- Odstranění přebytečných závorek a textu.
- Rozdělení řádku na dvojice souřadnic.
- Načtení souřadnic x a y a převedení souřadnic z datového typu *string* na *float*.
- Kontrola uložených souřadnic s x_{min} , x_{max} , y_{min} a y_{max} , případné přiřazení.
- Uložení bodů z řádku do polygonu.

Třída následně vrací načtené polygony a ohraničující souřadnice; tj. x_{min} , x_{max} , y_{min} a y_{max} ze všech načtených souřadnic souboru *csv*. Ve třídě *mainform* při otevření souboru *csv* se následně vypočte rozměr okna aplikace a rozměr načítaného souboru a ten se poté při vykreslování polygonů poměrově zmenší.

5 Vzhled aplikace

Grafické uživatelské rozhraní bylo vytvořeno pomocí sw. Qt Creator, jako Qt Widget Application. V horní části okna se nachází lišta s ovládacími prvky (načtení dat, generalizace, vyčištění plátna od kresby, konec). Všechny ovládací prvky jsou opatřeny nápovědou, která se zobrazí po krátkém umístění kurzoru na ikonu. Velikost okna aplikace je možné libovolně rozšiřovat.



Obrázek 2: Uživatelské rozhraní

6 Zhodnocení generalizačních metod

Zhodnocení zdali generalizační metoda správně či špatně vystihuje trend původního půdorysu budovy bylo vyhodnoceno čistě manuálně, jako průměr od tří nezávislých osob. Hodnoty v tabulkách 1a, 1b a 1c byly vypočteny dle vzorce (1). Na obrázcích 3, 4 a 5 je k nahlédnutí výběr "vhodných" a "špatných" aproximací.

$$\% = 100 \frac{\text{správně}}{\text{správně} + \text{špatně}} \quad (1)$$

typ zástavby	%
vilová čtvrť	98
sídliště	92
centrum	90

(a) MAER

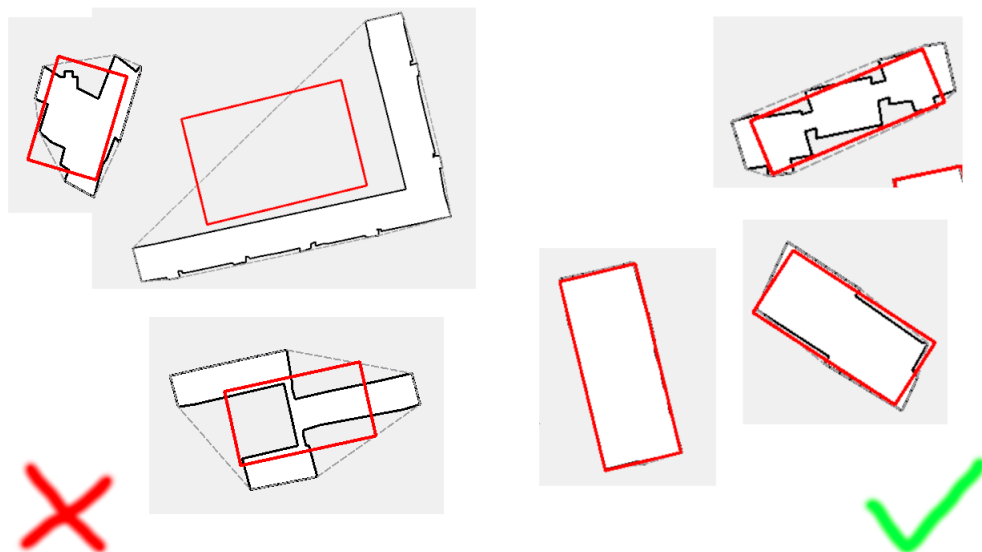
typ zástavby	%
vilová čtvrť	99
sídliště	86
centrum	91

(b) Wall Average

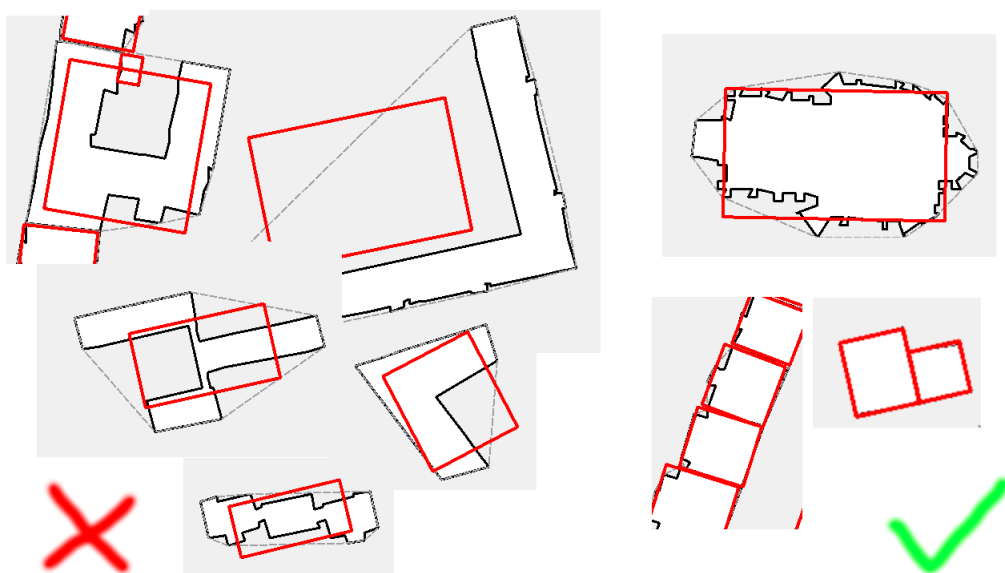
typ zástavby	%
vilová čtvrť	98
sídliště	85
centrum	87

(c) Longest Edge

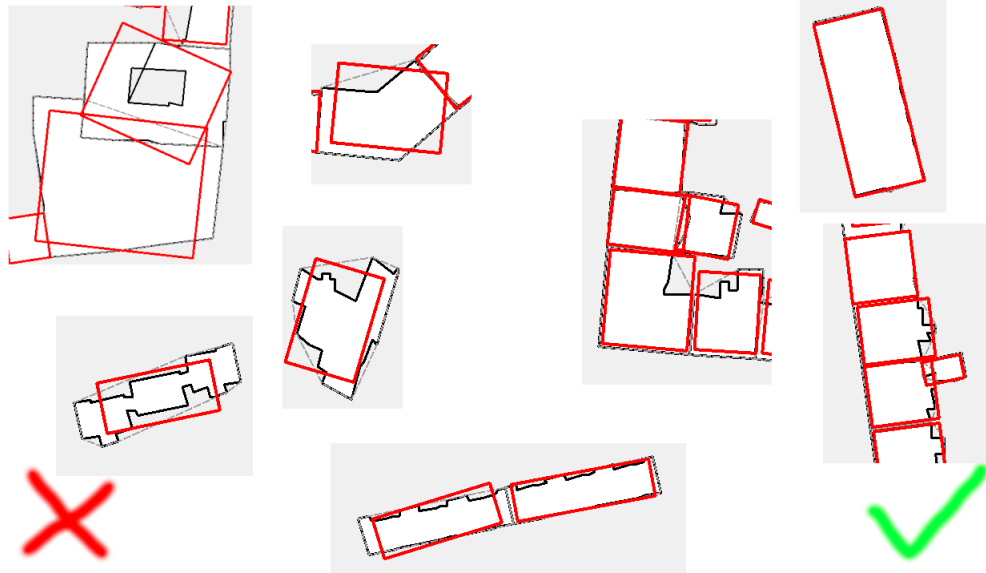
Tabulka 1: Procentuální úspěšnost jednotlivých metod



Obrázek 3: MAER



Obrázek 4: Wall Average



Obrázek 5: Longest Edge

7 Dokumentace

7.1 Třída Algorithms

- `int getPointAndLinePosition(QPointF &a, QPointF &p1, QPointF &p2);`
- `double get2LinesAngle(QPointF &p1, QPointF &p2, QPointF &p3, QPointF &p4);`
- `QPolygonF cHull(QPolygonF &b);`
- `QPolygonF rotate(QPolygonF &pol, double sigma);`
- `double area(QPolygonF &pol);`
- `tuple<double, QPolygonF> minMaxBox(QPolygonF &pol);`
- `QPolygonF minAreaEnclosingRect(QPolygonF &b);`
- `QPolygonF resRect(QPolygonF &rect, QPolygonF &b);`
- `QPolygonF simplifyAreaEnclosingRect(QPolygonF &b);`
- `QPolygonF simplifyWallAverage(QPolygonF &b);`
- `QPolygonF simplifyLongestEdge(QPolygonF &b);`

7.2 Třída Draw

- **explicit Draw**(QWidget *parent = nullptr);
- **void paintEvent**(QPaintEvent *event);
- **void setSimplifiedBuilding**(QPolygonF &bs_);
- **void setConvexBuilding**(std::vector(QPolygonF) &ch_)ch = ch_;
- **void clearAll**();
- **void clearRes**();
- **void drawPolygons**(std::vector(QPolygonF) &polygons, double &x_t, double &y_t, double &x_m, double &y_m);
- **QPolygonF transPolygon**(QPolygonF &polygon, double &x_t, double &y_t, double &x_m, double &y_m);

7.3 Třída WKT

- **vector(QPolygonF) readFile**(std::string &filename, double &xmin, double &xmax, double &ymin, double &ymax);

7.4 Třída MainForm : public QMainWindow

- **void on_actionOpen_triggered**();
- **void on_actionMin_Area_Rectangle_triggered**();
- **void on_actionWall_Average_triggered**();
- **void on_actionClear_All_triggered**();
- **void on_actionClear_results_triggered**();
- **void on_actionExit_triggered**();
- **void on_actionLongest_Edge_triggered**();

7.5 Třída sortPointsByX a třída sortPointsByY

- **bool operator**()(QPointF &p1, QPointF &p2);

8 Závěr

V rámci druhé úlohy předmětu Algoritmy digitální kartografie a GIS byla vytvořena aplikace nesoucí název BuildingSimplify. Aplikace byla napsána v jazyce C++ ve vývojovém prostředí Qt Creator.

Aplikace umožňuje načítání vektorových dat ve formátu WKT, jejich vykreslení a následnou generalizaci. Na výběr je celkem ze tří generalizačních metod (Minimum Area Enclosing Rectangle, Wall Average a Longest Edge).

Výsledná aplikace je k nalezení na této webové adrese: https://github.com/jaksno/adk_2023_24.

Reference

- [1] BAYER Tomáš, *Konverzní obálka množiny bodů*, [online]. [cit. 2023-11-13]. Dostupné z: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4_new.pdf
- [2] ISO 19125-1:2004, *Geographic information - Simple feature access*. [cit. 2023-11-13]. Dostupné z: <https://www.iso.org/standard/40114.html>