



ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

Úloha č. 2: **Generalizace budov**

Bc. Taťána Bláhová, Bc. Tomáš Krauz, Bc. Adéla Kučerová

28. října 2022

Obsah

1	Zadání	2
2	Bonusové úlohy	2
3	Popis a rozbor problému	3
4	Popis použitých algoritmů	3
4.1	Hledání konvexní obálky	3
4.2	Generalizace budov	3
5	Problematické situace a jejich rozbor	5
6	Vzhled aplikace	5
7	Vstupní data a výstupní	6
8	Dokumentace	9
8.1	Třída Algorithms	9
8.2	Třída Draw	9
8.3	Třída CSV	10
8.4	Třída Mainform	10
8.5	Třída SortPointsByX	10
8.6	Třída SortPointsByY	11
9	Závěr	11

1 Zadání

Úloha č. 2: Generalizace budov

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. 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 nahrad'te 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 zhodno'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 Weighted Bisector.	+8b
Implementace další metody konstrukce konverzní obálky.	+5b
Ošetření singulárního případu u při generování konverzní obálky.	+2b
Max celkem:	35b

Obrázek 1.1: Zadání úlohy

2 Bonusové úlohy

1. Generalizace budov metodou Longest Edge. [+5b]
2. Generalizace budov metodou Weighted Bisector. [+8]

3 Popis a rozbor problému

Na vstupu je množina polygonů (budov) $B = \{B_i\}_{i=1}^n$, kde budova $B_i = \{p[x, y]_{i,j}\}_{j=1}^m$.

Generalizace budovy, tj. zjednodušení tvaru polygonu za účelem redukce množství dat, je provedena pomocí konvexní obálky, jako první odhad tvaru prostorového jevu. Konvexní obálka je definována jako nejmenší konvexní mnohoúhelník P obsahující S (tj. neexistuje $P' \subset P$ splňující definici) množiny H konečné množiny S v E^2 . Množina S je označena jako konvexní, pokud spojnice libovolných dvou prvků leží zcela uvnitř této množiny. Jedná se tedy o ohraničující mnohoúhelník P s nejmenší plochou.

Generalizace je provedena pro každou budovu.

4 Popis použitých algoritmů

4.1 Hledání konvexní obálky

K hledání konvexní obálky byla využita metoda **Jarvis scan**.

Jako předzpracování je nalezen pivot q jako $q = \min_{p_i \in S}(y_i)$. Takto nalezený bod q je přidán do konvexní obálky H . Následně je vybrán bod p_{j-1} pro vytvoření přímky rovnoběžné s osou X . Přímka je daná body q a p_{j-1} .

Pomocí cyklu je přidán do konvexní obálky bod s maximálním úhlem $\angle(p_{j-1}, p_j, p_{j+1})$.

Postup je popsán pomocí vzorců na stránkách [1] .

Jarvis scan - implementace :

1. Nalezení pivotu q , $q = \min(y_i)$,
2. Přidání $q \rightarrow H$,
3. Inicializace $p_{j-1} \in X, p_j = q, p_{j+1} = p_{j-1}$,
4. Opakování, dokud $p_{j+1} \neq q$:
5. Nalezení $p_{j+1} = \argmax_{p_i \in P} \angle(p_{j-1}, p_j, p_i)$
6. Přidání $p_{j+1} \rightarrow H$
7. $p_{j-1} = p_j; p_j = p_{j+1}$.

4.2 Generalizace budov

4.2.1 Metoda Minimum Area Enclosing Rectangle

Pomocí této metody je zjištěn hlavní směr budovy. Směr je určen jako směr delší ze stran ohraničujícího obdélníku s minimální plochou.

Minimum Area Enclosing Rectangle - implementace :

1. Nalezení $H = CH(S)$
2. Inicializace $R = MMB(S)$, $\underline{A} = A(MMB(S))$
3. Opakování pro každou hranu e obálky H :
4. Spočtení směrnice σ hrany e ,
5. Otočení S o úhel $-\sigma$: $S_r = R(-\sigma)S$
6. Nalezení $MMB(S_r)$ a určení $A(MMB(S_r))$
7. Pokud $A < \underline{A}$:
8. $\underline{A} = A$, $\underline{MMB} = MMB$, $\underline{\sigma} = \sigma$
9. $R = R(\underline{\sigma})\underline{MMB}$

4.2.2 Metoda Longest Edge

Jedná se o detekci hlavního směru budovy, tj. nejdelší hrana polygonu, který budovu reprezentuje. Druhý hlavní směr je na ni kolmý.

Neplatí, že hlavní strana reprezentuje hlavní směr.

Metoda nedosahuje nejlepších výsledků při netypických tvarech polygonu.

Longest Edge - implementace :

1. Inicializace vektoru dvojic délka hrany a směrnice přímky
2. Opakování pro všechny body polygonu:
- 3.

$$s_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (4.1)$$

4.

$$\sigma_j = \arctan 2 \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \quad (4.2)$$

5. Seřazení dvojic podle velikosti s_j
6. Uložení $\sigma_{s,max}$, tj. poslední prvek ve vektoru dvojic
7. Vytvoření enclosing rectangle (ohraničující obdélník)

4.2.3 Metoda Wall Average

Všem stranám polygonu (budovy) je spočtena směrnice σ_i , na kterou je aplikována metoda $mod(\frac{\Phi}{2})$ pro nalezení zbytku po dělení. Výsledný směr natočení polygonu je pak dán váženým průměrem těchto zbytků, kde roli váhy zastupuje délka příslušné strany.

Metoda je komplexní a citlivá na úhly různé od úhlů pravých.

Wall Average - implementace :

1. Inicializace pro $\sigma = 0$... směr natočení budovy;

2. $\Sigma s_i = 0$... obvod budovy;
3. σ ... směrnice první hrany budovy
4. Opakování přes všechny body polygonu:
- 5.

$$s_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (4.3)$$

$$\sigma_j = \arctan 2 \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \quad (4.4)$$

$$d\sigma_i = \sigma_i - \sigma \quad (4.5)$$

$$k_i = \text{round} \frac{2 * d\sigma_i}{\pi} \quad (4.6)$$

$$r_i = d\sigma_i - k_i * \frac{\pi}{2} \quad (4.7)$$

$$\sigma+ = r_i * s_i \quad (4.8)$$

$$\Sigma s_i+ = s_i \quad (4.9)$$

$$\sigma = \sigma + \frac{\sigma}{\Sigma s_i} - \text{váž.průměr} \quad (4.10)$$

13. Vytvoření enclosing rectangle (ohraničující obdélník).

5 Problematické situace a jejich rozbor

Při tvorbě aplikace se nepodařilo spustit metody pro načtený soubor dat. Aplikace funguje pouze pro manuálně naklikaný polygon.

6 Vzhled aplikace

Spuštěním aplikace se otevře okno "Building Simplify". Okno je rozděleno na dvě části. Na levé straně se nachází Canvas, část vyhrazená pro vykreslování geometrie. Vpravo je menu, které obsahuje tlačítka tříd QPushButton a rolovací menu QComboBox.



Obrázek 6.1: Vzhled aplikace

Tlačítka QPushButton slouží k nahrání dat, generalizaci, pročištění grafického okna (Canvas). Rolovací menu slouží k výběru metody generalizace.

7 Vstupní data a výstupní

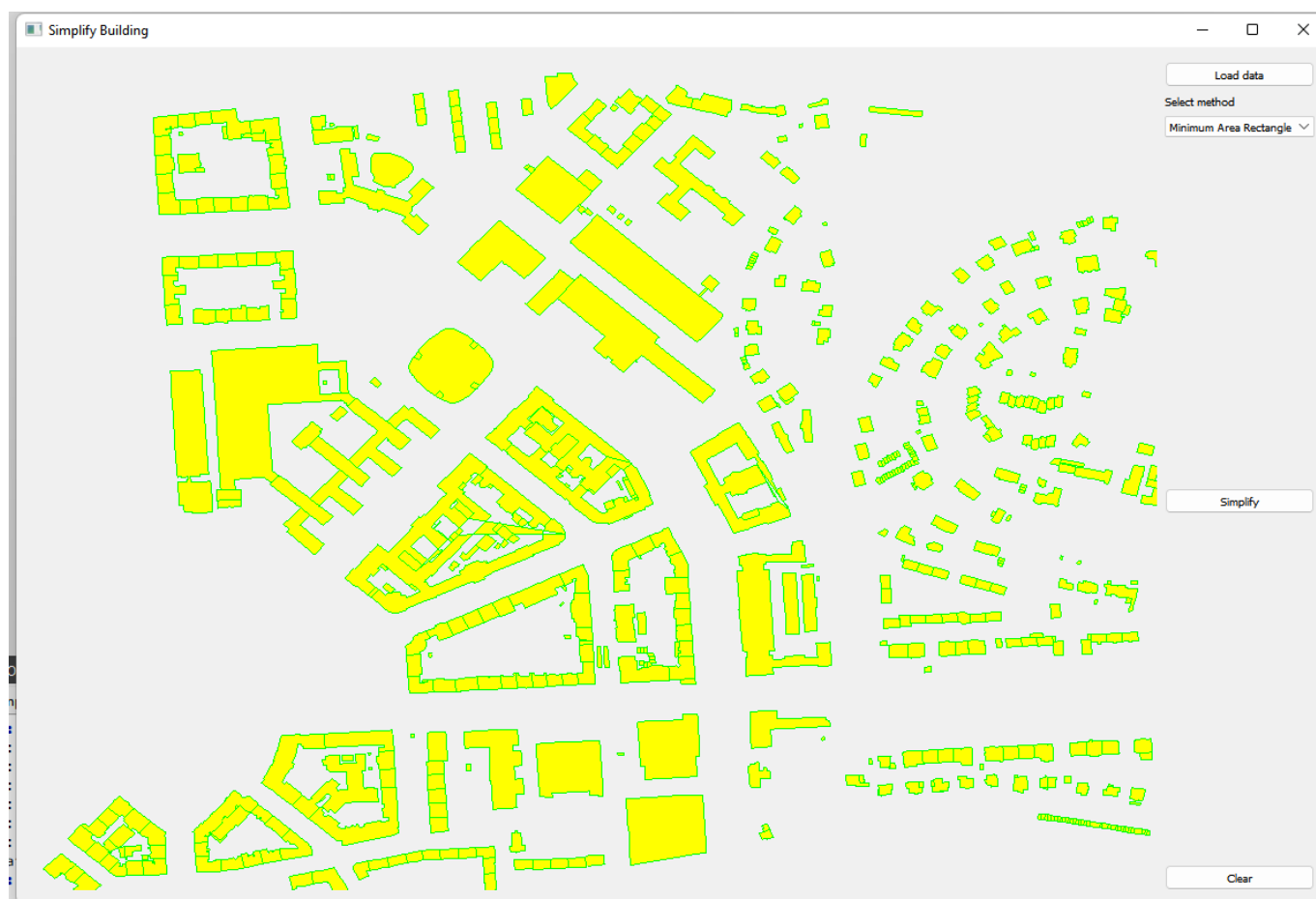
Pomocí tlačítka Load file lze do aplikace nahrát soubor polygonů ve formátu CSV. Ukázková data jsou polygony budov v oblasti Praha 6, Dejvice. Souřadnice jsou v S-JTSK.

1	WKT,									
2	MULTIPOLYGON	(((744694.72 1040883.16,744692.03 1040887.16,744692.028 1040887.164,744690.865 1040889.096								
3	MULTIPOLYGON	(((744757.29 1040874.81,744755.72 1040873.31,744755.716 1040873.307,744752.32 1040870.28,74								
4	MULTIPOLYGON	(((745068.93 1040947.45,745066.3 1040943.03,745064.99 1040943.99,745059.926 1040935.651,745								

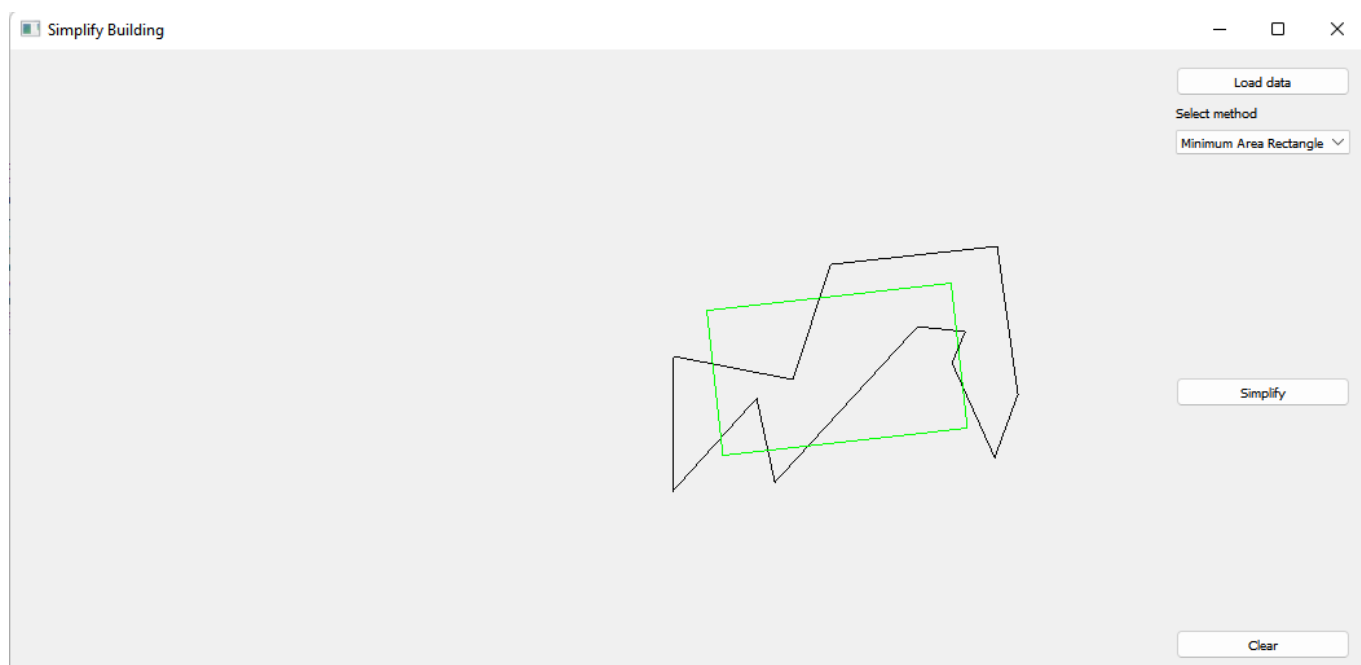
Obrázek 7.1: Vstupní data

Výstupem aplikace je grafické znázornění nahraných objektů (polygonů) a příslušná generalizace pro individuálně naklikaný polygon.

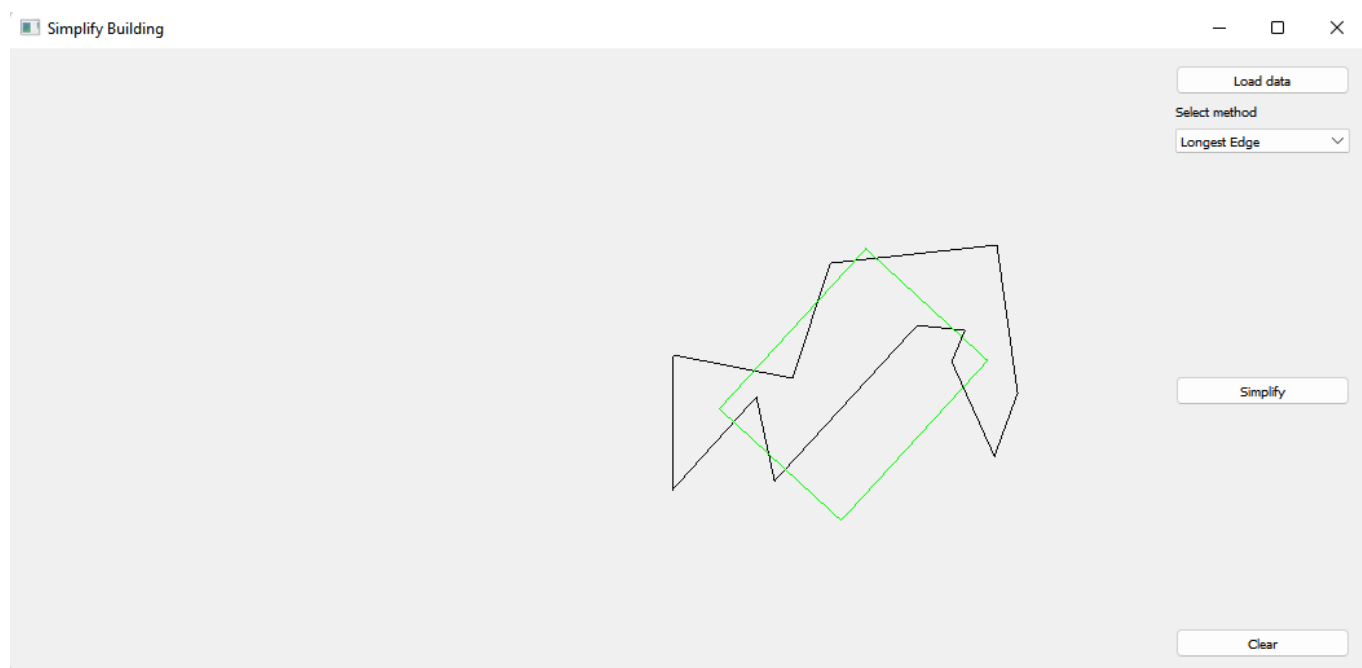
Výsledky jsou ukázány pro jeden polygon.



Obrázek 7.2: Výstupní data



Obrázek 7.3: Výstupní data



Obrázek 7.4: Výstupní data



Obrázek 7.5: Výstupní data

8 Dokumentace

8.1 Třída Algorithms

- *int getPointLinePosition(QPointF &p1, QPointF &p2, QPointF &q)*
 - analyzuje vzájemný vztah bodu a přímky
- *double getTwoLinesAngle(QPointF &p1, QPointF &p2, QPointF &p3, QPointF &p4)*
 - vypočítá úhel mezi dvěma vektory
- *QPolygonF createCH(QPolygonF &pol)*
 - vypočítá konvexní obálku polygonu
- *QPolygonF rotate(QPolygonF &pol, double sig)*
 - pootočí množinu o úhel
- *double getArea(QPolygonF &pol)*
 - vypočítá plochu polygonu
- *tuple < QPolygonF, double > minMaxBox(QPolygonF &pol)*
 - vytvoří MinMaxBox
- *QPolygonF minAreaEnclosingRectangle(QPolygonF &pol)*
 - vytvoří minimální ohraničující obdélník
- *QPolygonF resizeRectangle(QPolygonF &rec, double areaB)*
 - změní velikost obdélníku
- *QPolygonF resizeMinAreaEnclosingRectangle(QPolygonF &pol)*
 - změní velikost minimálního ohraničujícího obdélníku
- *QPolygonF wallAverage(QPolygonF &pol)*
 - provede metodu Wall Average
- *QPolygonF longestEdge(QPolygonF &pol)*
 - provede metodu Longest Edge

8.2 Třída Draw

- *void mousePressEvent(QMouseEvent * event)*
 - vrátí souřadnice kurzoru po kliknutí na Canvas
- *void paintEvent(QPaintEvent * event)*
 - vykreslí polygony na Canvas
- *QPolygonF getCH(){return ch;}*
 - vykreslí konvexní obálku Convex Hull
- *QPolygonF getMAER(){return er;}*

- vykreslí Minimum Area Enclosing Rectangle
- *QPolygonF getBuild()return building;*
 - vykreslí polygony na Canvas
- *void setCH(QPolygonF &ch){ch = ch;}*
 - vrátí Convex Hull
- *void setMinimumAreaEnclosingRectangle(QPolygonF &er){er = er;}*
 - vrátí Minimální ohraničující obdélník
- *void clearAll();*
 - vyčistí grafické okno
- *QPolygonF transformPolygon(QPolygonF &pol, double&x_tans, double&y_tans, double &x_ratio, double &y_ratio);*
 - transformuje polygon MinMax boxu
- *void drawPolygons(std :: vector < QPolygonF > &pol, double &x_tans, double &y_tans, double &x_ratio, double &y_ratio)*
 - vykreslí polygony na Canvas
- *vector < QPolygonF > getPolygons() polygons; }*
 - vrátí polygon

8.3 Třída CSV

- *vector < QPolygonF > read_CSV(string &filename)*
 - načte vstupní csv soubor

8.4 Třída Mainform

- *void on_simplify_clicked()*
 - provedení vybrané metody
- *void on_load_data_clicked()*
 - otevře průzkumníka souborů
- *void on__clear_clicked()*
 - vyčistí Canvas

8.5 Třída SortPointsByX

- *bool operator () (QPointF &p, QPointF&q)*
 - seřadí body podle velikosti souřadnice X

8.6 Třída SortPointsByY

- *bool operator () (QPointF &p, QPointF&q)*
 - seřadí body podle velikosti souřadnice Y

9 Závěr

Byla vytvořena aplikace Building Simplify s grafickým rozhráním. Aplikace byla napsána v programovacím jazyce C++ a umožňuje nahrání souboru CSV. Měla fungovat tak, že by bylo následně možné provést simplifikaci polygonů dle zvolené metody (Minimum Area Enclosing Rectangle, Wall Average, Longest Edge). Aplikaci je možné spustit, nahrát do ní soubor csv nebo je možné si polygon vlastnoručně naklikat, ale metody je možné spustit pouze pro individuálně naklikaný polygon.

Aplikaci jednotlivých metod hodnotíme následovně:

Minimum Bounding Rectangle

- Natočení budovy je dáno delší hranou MBB. Metoda dává dobré výsledky, je výpočetně náročnější. Vstupní množina bodů se pro aplikaci metody musí znovu transformovat v každé iteraci.

Wall Average

- Natočení budovy je dáno směrem nejdelší hrany. Výpočet není tolik náročný. Výsledky jsou adekvátní, ale nejdelší strana nemusí vypovídat o natočení celé budovy

Longest Edge

- Každé směrnici je vypočten celočíselný zbytek po dělení hodnotou $\frac{\pi}{2}$. Metoda dává dobré výsledky, ale pro každou směrnici je nutno spočítat modus.

Literatura

- [1] Tomáš Bayer, *Konvexní obálka množiny bodů*, <https://agony.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf>, [28.10.2022].