

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

KATEDRA GEOMATIKY

Úloha č. 2: Konvexní obálky

Monika KŘÍŽOVÁ
Marek HOFFMANN

06.11.2021

Obsah

1	Zadání	2
2	Údaje o bonusových úlohách	4
3	Popis problému	4
4	Popisy algoritmů	5
4.1	Tvorba konvexní obálky	5
4.1.1	Jarvis scan	5
4.1.2	Quick Hull	5
4.2	Generalizace budov	5
4.2.1	Minimum Enlosing Rectangle	5
4.2.2	Metoda Wall Average	7
4.2.3	Metoda Longest Edge	7
4.2.4	Metoda Weighted Bisector	8
5	Problematické situace	8
6	Vstupní data	8
7	Výstupní data	8
8	Dokumentace	8
8.1	Třída Algorithms	8
9	Závěr	9

1 Zadání

Vytvořte aplikaci s grafickým uživatelským rozhraním, která bude ze souboru načítat polygony budov a následně je bude generalizovat na obdélníky s plochou odpovídající generalizované budově a bude vytvářet konvexní obálky. Hlavní směry budov budou určeny metodami Minimum Area Enclosing Rectangle a Wall Average.

Přesné zadání je vloženo na následující stranu.

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 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 Weighted Bisector	+8b
Implementace další metody konstrukce konvexní obálky.	+5b
Ošetření singulárního případu u při generování konvexní obálky	+2b
Max celkem:	35b

Čas zpracování: 3 týdny.

2 Údaje o bonusových úlohách

Zpracovány byly celkem 3 bonusové úlohy ze 4 zadanych.

- Generalizace budov metodou Longest Edge +5b
- Generalizace budov metodou Weighted Bisector +8b
- Implementace další metody konstrukce konvexní obálky. +5b

Generalizace budov metodou Longest Edge byla implementována ve třídě Algorithms, je možné ji spustit pomocí tlačítka Building simplify po předchozím výběru v comboBoxu umožňujícího výběr metody generalizace polygonu.

Generalizace budov metodou Weighted Bisector byla stejně jako metoda Longest Edge implementována ve třídě Algorithms a taktéž je ji možné spustit pomocí tlačítka Building simplify po předchozím výběru v comboBoxu umožňujícího výběr metody generalizace polygonu.

Pro implementaci další metody konstrukce konvexní obálky byla vybrána metoda Quick Hull, jež byla přidána do třídy Algorithms. Pro její spuštění je nutné vybrat tuto metodu v náležitém comboBoxu a následně stisknout tlačítko Create convex hull.

3 Popis problému

Generalizace je v kartografii velmi důležitý proces, který je používán například při změně měřítka mapy, změně jejího účelu nebo pro zlepšení grafické přehlednosti. Generalizace může být provedena několika různými metodami, v řešené úloze byla aplikována tzv. geometrická generalizace, jež spočívá ve zjednodušení tvaru objektu. Objekty (v této úloze budovy) byly generalizovány na obdélníky s plochou odpovídající ploše původních objektů.

Jednotlivé generalizační algoritmy hledají hlavní směr natočení budovy, po nalezení tohoto směru se vytvoří opsaný obdélník s minimálním obsahem, jež je natočený do tohoto směru (dále pod zkratkou MMB). MMB se následně zmenší tak, aby byla jeho plocha totožná s plochou generalizovaného objektu.

4 Popisy algoritmů

4.1 Tvorba konvexní obálky

4.1.1 Jarvis scan

Prvním krokem při tvorbě konvexní obálky \mathcal{H} metodou Jarvis scan je nalezení pivotu q . Tento bod je bod s nejmenší souřadnicí y v množině bodů S . Dalším krokem je nalezení bodu p , jež svírá největší úhel s rovnoběžkou s osou x procházející bodem q . Nalezený bod se přidá do \mathcal{H} a opět se hledá bod, jež bude svírat s předchozím bodem největší úhel. Tímto způsobem se do \mathcal{H} přidávají body do okamžiku, kdy by byl přidávaným bodem bod q . Tato metoda je velmi jednoduchá, ale kvůli časové náročnosti nevhodná pro velké množiny bodů.

4.1.2 Quick Hull

V algoritmu Quick Hull je nejprve nutné seřadit množinu bodů S podle souřadnice x a následně ze seřazeného souboru vybrat dva body, tzv. pivoty. Bod q_1 je bodem s nejmenší souřadnicí x , bod q_3 má naopak souřadnici x ze souboru bodů S největší. Konvexní obálka \mathcal{H} se vytváří ze dvou částí – horní (\mathcal{H}_U) a spodní \mathcal{H}_L konvexní obálky. Do \mathcal{H}_U se přidávají body ležící nalevo od spojnice pivotů q_1 a q_3 , do \mathcal{H}_L naopak body ležící na pravé straně od spojnice těchto bodů.

Posledním krokem je spojení obou částí konvexní obálky do výsledné množiny \mathcal{H} . Nejprve se do \mathcal{H} přidá bod q_3 , poté se provede rekurze \mathcal{H}_U , pomocí níž se do \mathcal{H} přidají body z horní části konvexní obálky, poté se přidá bod q_1 a posledním krokem je provedení rekurze \mathcal{H}_L , díky níž se do konvexní obálky přidají i body ze spodní části konvexní obálky.

4.2 Generalizace budov

4.2.1 Minimum Enlosing Rectangle

Algoritmus se snaží vyhledat takovou hranu konvexní obálky \mathcal{H} , aby po vytvoření opsaného obdélníka s jednou stranou kolineární s touto hranou měl takto vytvořený obdélník minimální plochu.

Algoritmus využívá již zkonstruovanou konvexní obálku \mathcal{H} , pro jejíž hrany jsou vypočítávány směrnice σ hrany e :

$$\tan \sigma = \frac{d_x}{d_y}, \quad (1)$$

kde d_x a d_y jsou souřadnicové rozdíly počátečního a koncového bodu hrany konvexní obálky \mathcal{H} . Všechny body množiny S se následně pomocí matice rotace R otočí o úhel $-\sigma$:

$$S_R = R(-\sigma)S \quad (2)$$

Pro otočené body množiny se vytvoří MMB s následujícími souřadnicemi vrcholů:

$$V_1 = [\underline{x}, \underline{y}], V_2 = [\bar{x}, \underline{y}], V_3 = [\bar{x}, \bar{y}], V_4 = [\underline{x}, \bar{y}], \quad (3)$$

kde $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ jsou minimální a maximální souřadnice natočených bodů množiny S_R . Vypočte se plocha vytvořeného MMB:

$$A = (\bar{x} - \underline{x})(\bar{y} - \underline{y}), \quad (4)$$

jež se následně porovná s minimální uloženou plochou. Je-li vypočtená plocha A menší než je plocha minimální, uloží se jako nové minimum. Dále se uloží pro tuto plochu uloží úhel σ_{min} a MMB_{min} . Po ukončení výpočtu tohoto cyklu je již vypočten výsledný úhel natočení budovy σ_{min} . Následně je nutné MMB natočit o tento úhel a zmenšit tak, aby byla plocha MMB rovna ploše generalizovaného polygonu. Tento postup bude používán i pro další metody.

MMB_{min} se otočí o úhel σ_{min} :

$$\mathcal{R} = R(\sigma_{min})MMB_{min}, \quad (5)$$

následně se jeho plocha proporcionálně zmenší vůči těžišti tak, aby byla rovna ploše generalizovaného polygonu. Nejprve je nutné vypočítat poměr k plochy A_b generalizované budovy a plochy A MMB

$$k = \frac{A_b}{A}. \quad (6)$$

Souřadnice těžiště T jsou aritmetickým průměrem souřadnic vrcholů \mathcal{R} . Vypočítají se nové vrcholy V obdélníku \mathcal{R}_r :

$$V_i = T + \sqrt{k}u_i, \quad (7)$$

kde u_i jsou směrové vektory vrcholů a těžiště odvozené z Pythagorovy věty:

$$u_i = V_i - T \quad (8)$$

4.2.2 Metoda Wall Average

Metoda je velmi komplexní a citlivá na nepravé úhly. Pro každou hranu je pro výsledky operace $\text{mod}()$ vypočítáván vážený průměr, v němž je vahou délka hrany.

Pro každou hranu budovy se zredukuje směrnice σ :

$$\Delta\sigma = \sigma_i - \sigma', \quad (9)$$

kde σ_i je směrnice hrany vypočtená dle vztahu (1) a σ' je směrnice první hrany vypočtená dle vztahu (1).

V dalším kroku se vypočítá zaokrouhlený podíl:

$$k_i = \left\lfloor \frac{2\Delta\sigma_i}{\pi} \right\rfloor \quad (10)$$

a následně se dopočítá zbytek po dělení:

$$r_i = \Delta\sigma_i - k_i \frac{\pi}{2}. \quad (11)$$

Výsledný směr natočení budovy roven:

$$\sigma = \sigma' + \sum \frac{r_i s_i}{s_i}. \quad (12)$$

Následně se vytvoří MMB, který se otočí o úhel σ a zmenší do požadované plochy. Tento postup byl již shrnut v kapitole 4.2.1.

4.2.3 Metoda Longest Edge

Hlavní směrem budovy se dle této metody rozumí směrnice nejdelší hrany budovy, druhý směr je kolmý na směr hlavní. Pomocí cyklu se postupně vypočítává délka d všech hran generalizované budovy, postupně se zjišťuje, je-li vypočtená délka větší než je dosavadně největší uložená délka.

$$d = \sqrt{d_x^2 + d_y^2} \quad (13)$$

Následně se vytvoří MMB, který se otočí o úhel σ a zmenší do požadované plochy. Tento postup byl již shrnut v kapitole 4.2.1.

4.2.4 Metoda Weighted Bisector

Algoritmus hledá dvě nejdelší úhlopříčky generalizované budovy, pro tyto dvě úhlopříčky se následně vypočítají směrnice (rovnice (1)) a délky hran (rovnice (13)). Výsledný směr se získá z váženého průměru:

$$\sigma = \frac{s_1\sigma_1 + s_2\sigma_2}{s_1 + s_2}. \quad (14)$$

Následně se vytvoří MMB, který se otočí o úhel σ a zmenší do požadované plochy. Tento postup byl již shrnut v kapitole 4.2.1.

5 Problematické situace

Během práce na úloze se vyskytly problémy při načítání dat, které byly uloženy v souřadnících S-JTSK, z něhož je bylo nutné transformovat do souřadnic plátna pro vykreslování.

Prvním krokem byla změna měřítka souřadnic a zaměnění souřadnice x za y a naopak, následovalo získání minimálních a maximálních souřadnic x a y bodů množiny S . Z vypočtených minim a maxim se následně vypočítal rozsah mezi těmito souřadnicemi, kterým se vydělil rozsah plátna, čímž se získalo měřítko, jímž se následně souřadnice vynásobily.

6 Vstupní data

7 Výstupní data

8 Dokumentace

8.1 Třída Algorithms

QPolygonF cHull (std::vector <QPointF> &points)

Funkce vytváří konvexní obálku metodou Jarvis scan popsanou v kapitole 4.1.1. Vstupním argumentem funkce je vektor obsahující body množiny S , body jsou uloženy jako *QPointF*. Návratovým typem funkce je *QPolygonF*, tedy polygon se souřadnicemi uloženými s přesností float obsahující souřadnice vrcholů vypočtené konvexní obálky.

QPolygonF qHull (std::vector<QPointF> &points)

Funkce vytváří konvexní obálku metodou Quick Hull popsanou v kapitole 4.1.2. Vstupním argumentem funkce je vektor obsahující body množiny S , které jsou uloženy jako *QPointF*. Návrátovým typem funkce je *QPolygonF*, tedy polygon se souřadnicemi uloženými s přesností float, který obsahuje souřadnice vrcholů vypočtené konvexní obálky.

void qHullRecursive(int r, int s, std::vector<QPointF> &points, QPolygonF &ch)

Funkce je rekurzivní funkcí k výše uvedené funkci qHull. Touto funkcí se přidávají části konvexní obálky \mathcal{H}_U a \mathcal{H}_L do výsledné konvexní obálky \mathcal{H} . Vstupními argumenty jsou integery r a s , jež označují směr linie mezi pivoty q_1 a q_3 . Tyto body jsou uloženy ve vektoru *points* (popsán níže) na prvních dvou místech, volá-li se tedy funkce například s těmito parametry: qHullRecursive(1,0,su,qh), bude počátečním bodem linie pivot q_3 a konečným bodem pivot q_1 . Třetím vstupním argumentem je vektor bodů *QPointF points*, do této proměnné se ukládá \mathcal{H}_U nebo \mathcal{H}_L . Posledním vstupním argumentem je *QPolygonF*, do nějž se bude ukládat horní/spodní část konvexní obálky. Návrátový typ funkce je void, funkce tedy nevrací žádnou proměnnou, pouze ukládá body do předem vytvořené proměnné.

9 Závěr