

České vysoké učení technické v Praze
Fakulta stavební



155ADKG Algoritmy v digitální kartografii

Konvexné obálky a ich konštrukcie

Bc. Lukáš Kettner Bc. Martin Hulín
3.11.2019

Obsah

1	Zadanie	2
1.1	Bonusové úlohy	2
2	Popis a rozbor problému	2
3	Popis použitých algoritmov	3
3.1	Jarvis Scan	3
3.1.1	Implementácia metódy	3
3.1.2	Problematické situácie	4
3.2	Quick Hull	4
3.2.1	Implementácia metódy	4
3.2.2	Problematické situácie	4
3.3	Sweep Line	4
3.3.1	Implementácia metódy	5
3.3.2	Problematické situácie	6
3.4	Graham scan	6
3.4.1	Implementácia metódy	6
4	Vstupné dáta	6
4.1	Množina bodov	6
4.2	Množina bodov vytvorená ručne	7
4.3	Automaticky generovaná množina bodov	7
5	Ukážka vytvorenej aplikácie	7
6	Vykreslenie konvexnej obálky	7
7	Dokumentácia	7
7.1	Trieda Algorithms	7
7.1.1	Metódy	7
7.2	Trieda Draw	8
7.2.1	Členské premenné	8
7.2.2	Metódy	8
7.3	Trieda Widget	8
7.3.1	Metódy	8
8	Záver	9

1 Zadanie

Vytvorte aplikáciu s grafickým rozhraním, ktorá vygeneruje konvexnú obálku podľa zvoleného typu algoritmu. Vstup do aplikácie : množina bodov $\{p_1, \dots, p_n\}$. Výstup aplikácie : konvexná obálka $H(P)$.

Nad množinou P implementujte nasledujúce algoritmy pre konštrukciu $H(P)$.

- Jarvis Scan
- Quick Hull
- Sweep Line

Vstupné množiny vrátane vygenerovaných konvexných obálok vhodne vizualizujte. Vytvorte grafy pre množiny $n \in \langle 1000, 1000\ 000 \rangle$ ilustrujúce doby behu algoritmu. Meranie prevádzajte pre rôzne typy množín opakovane 10x a uveďte rozptyl. Namerané údaje usporiadajte do tabuliek.

Taktiež sa zamyslite nad problémom singularít pre rôzne typy vstupných množín a nad možnými optimalizáciami. Zhodnoťte dosiahnuté výsledky. Rozhodnite, ktorá z týchto metód je vzhľadom na časovú náročnosť a typ vstupnej množiny najvhodnejšia.

1.1 Bonusové úlohy

V rámci úlohy sú vypracované tieto bonusové úlohy

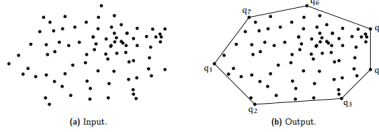
- Konštrukcia konvexnej obálky metódou Graham Scan.

2 Popis a rozbor problému

Konvexná obálka je skupina bodov, ktorých spojením vznikne ohraničenie pre všetky ostatné body množiny. Pre konvexnú obálku platí :

- žiadny bod vstupnej množiny P neleží mimo ohraničenia konvexnej obálky.
- všetky spojnice bodov vstupnej množiny P ležia vnútri konvexnej obálky alebo tvoria jej ohraničenie.

- všetky vnútorné uhly medzi susednými segmentami konvexnej obálky sú menšie ako 180 stupňov.



Obr. 1: Princíp konvexnej obálky

Konvexnú obálku je možné zostrojiť viacerými metódami. V rámci našej práce sme zostrojili algoritmus Jarvis Scan, Quick Hull, Sweep Line a Graham scan.

3 Popis použitých algoritmov

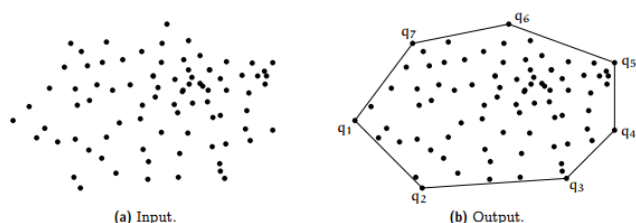
3.1 Jarvis Scan

Tento algoritmus predstavuje jeden z najpoužívanějších postupov pre tvorbu konvexnej obálky. V algoritme sa zavádza kritérium maximálneho uhlu (ω_{max}). Pri tomto kritériu posudzujeme uhol medzi poslednou stranou konvexnej obálky a úsečkou posledný bod obálky - aktuálny bod. Kritérium hľadá taký aktuálny bod, pre ktorý je uhol (ω) maximálny. Postup je nasledovný. V prvom kroku vyberieme pivot q so súradnicou y_{min} , tento bod je automaticky súčasťou konvexnej obálky. Zavádzame kritérium maximálneho uhlu (ω_{max}). Keď nájdeme bod odpovedajúci kritériu, pridáme takýto bod do konvexnej obálky. Algoritmus končí v momente keď sa dostaneme opäť k pivotu q .

3.1.1 Implementácia metódy

1. Nájdí pivot q . Zoraď body podľa súradnice y . $q : y_{min}$
2. Do konvexnej obálky pridaj q
3. Inicializuj $p_{j-1} \in X, p_j = q, p_{j+1} = p_{j-1}$
4. Opakuj pokiaľ $p_{j+1} \neq q$
5. Nájdienie bodu pre ktorý je uhol omega maximálny $p_{j+1} = \underset{p_i \in P}{argmax} \angle(p_{j-1}, p_j, p_i)$

6. Pridaj nájdený bod p_{j+1} do konvexnej obálky
7. Inicializuj $p_{j-1} = p_j, p_j = p_{j+1}$



Obr. 2: Princíp konvexnej obálky

3.1.2 Problematické situácie

Problematickou situáciou sú vyskytujúce sa kolineárne body. V takomto prípade je potrebné ošetriť aby sa do konvexnej obálky dostal najvzdialenejší bod.

3.2 Quick Hull

Tento algoritmus uplatňuje pri konštrukcii konvexnej obálky princíp rozdeľuj a panuj. Jedná sa o rekurzívny algoritmus. Skladá sa z lokálnej a globálnej procedúry. Konvexná obálka je skonštruovaná z dvoch častí - upper hull a lower hull. Deliacou priamkou je spojnica dvoch bodov s extrémnymi súradnicami x (minimum, maximum). S oboma časťami obálky sa počas výpočtu pracuje samostatne. Algoritmus hľadá bod s extrémnymi súradnicami v ose y , samostatne pre hornú a dolnú časť. Výsledná konvexná obálka je pri zachovaní CCW orientácie zložená z dvoch častí automaticky, bez nutnosti spájania.

3.2.1 Implementácia metódy

1. Tady něco bude

3.2.2 Problematické situácie

3.3 Sweep Line

Jedná sa o metódu zametacej priamky. Tento algoritmus pracuje na princípe prepisovania pozície predchodcu a následníka i -teho bodu. Popis algoritmu

slovne je miestami zložité pochopiť a pri jeho prezentácii pred osobou mimo obor by sme mohli skončiť na psychiatrii. Implementácia metódy jednoduchšie zobrazí jej fungovanie.

3.3.1 Implementácia metódy

1. Zoradenie množiny bodov P_s podľa súradnice x
2. if $p_3 \in \sigma_L(p_1, p_2)$
3. $n[1] = 2; n[2] = 3; n[3] = 1$
4. $p[1] = 3; p[2] = 1; p[3] = 2$
5. else
6. $n[1] = 3; n[3] = 2; n[2] = 1$
7. $p[1] = 2; p[3] = 1; p[2] = 3$
8. for $p_i \in P_s, i > 3$
9. if $(y_i > y_i - 1)$
10. $p[i] = i-1; n[i] = n[i-1]$
11. else
12. $n[i] = i-1; p[i] = p[i-1]$
13. $n[p[i]] = i; p[n[i]] = i;$
14. while $(n[n[i]]) \in \sigma_R(i, n[i])$
15. $p[n[n[i]]] = i; n[i] = n[n[i]];$
16. while $(p[p[i]]) \in \sigma_L(i, p[i])$
17. $n[p[p[i]]] = i; p[i] = p[p[i]];$

3.3.2 Problematické situácie

3.4 Graham scan

Grahamov prehľadávací algoritmus využíva kritérium pravotočivosti, pri ktorom posudzuje uhol (ω_i). Množina bodov je zoradená podľa súradnice y. Bod s najmenšou y súradnicou je zvolený za pivota. Z pivota vedieme rovnobežku s osou X. Následne určíme uhol od osi X k ostatným bodom množiny. Tieto uhly je potrebné zotiediť podľa veľkosti. Následne vyberieme bod (p_i) pre ktorý je uhol (ω_i) maximálny. Pri pridávaní bodu do konvexnej obálky musí byť splnená podmienka ľavotočivosti.

3.4.1 Implementácia metódy

1. Zoradenie množiny bodov P_s podľa súradnice y
2. Nájdenie pivota q, $q = \min(y_i)$, $q \rightarrow \text{Convex Hull}$
3. Zorad' body podľa veľkosti uhlu (ω_i) $\angle(p_{kolinear}byx, x, p_i)$
4. Vymaž bližší bod ku q *if* ($\omega_i = \omega_j$)
5. Opakuj pre všetky body j i n
6. Pridaj bod p_j pre ktorý je uhol omega maximálny a je splnená podmienka ľavotočivosti $\rightarrow \text{Convex Hull}$
7. $j = j+1$
8. Else pop S

4 Vstupné dáta

Vstupnými dátami je množina bodov. Táto množina môže byť ručne naklikaná v Canvase alebo automaticky generovaná.

4.1 Množina bodov

Typy zadania množiny bodov :

4.2 Množina bodov vytvorená ručne

Táto množina vznikne ručným naklikaním bodov priamo v Canvase.

4.3 Automaticky generovaná množina bodov

Množina vznikne automatickým vygenerovaním po zadaní požadovaného počtu generovaných bodov.

5 Ukážka vytvorenej aplikácie

6 Vykreslenie konvexnej obálky

7 Dokumentácia

7.1 Trieda Algorithms

Triedu Algorithms sme použili pre deklarovanie funkcií pre výpočtové algoritmy tvorby konvexnej obálky a pre deklaráciu ich pomocných metód.

7.1.1 Metódy

getAngle2Vectors

- je pomocnou metódou pre metódu positionPointPolygonWinding. Slúži k určeniu uhlu medzi dvoma priamkami. Jej návratovou hodnotou je double
- na vstupe má : súradnice bodov p_1, p_2, p_3, p_4 určujúcich prvú a druhú priamku
- výstupom je hodnota uhlu medzi priamkami

getPointLinePosition

- je pomocnou metódou pre metódu positionPointPolygonWinding. Slúži na určenie polohy bodu voči priamke. Jej návratovou hodnotou je integer.
- na vstupe má : súradnice určovaného bodu q , súradnice bodov priamky $p_1 p_2$

- na výstupe hodnoty :
 - 1 – bod sa nachádza na priamke
 - 0 – bod sa nachádza vpravo od priamky
 - 1 – bod sa nachádza vľavo od priamky

7.2 Trieda Draw

Trieda Draw slúži ku grafickému vykresleniu bodu q a konvexnej obálky.

7.2.1 Členské premenné

7.2.2 Metódy

paintEvent

- slúži k vykresleniu zisťovaného bodu, vykresleniu importu polygónovej mapy a k vyfarbeniu polygónov ktorým náleží určený bod q. Návrátovým typom je void.

void mousePressEvent

- slúži k vykresleniu bodu q stlačením tlačidla myši, v okamihu stlačenia tlačidla na myši sa uložia súradnice bodu q. Návrátovým typom je void.

void clearCanvas

- slúži k vymazaniu obsahu vykreslovacieho plátna. Návrátovým typom je void.

7.3 Trieda Widget

Trieda Widget obashuje metódy ktoré sú odkazom na sloty umožňujúce vykonávať príkazy z grafického rozhrania aplikácie. Nemajú žiadne vstupné hodnoty, návratovým typom je void.

7.3.1 Metódy

on_clearButton_clicked

- tlačidlo Clear, vyčistí grafické okno Canvasu

8 Závěr