České vysoké učení technické v Praze
Fakulta stavební

Algoritmy v digitální kartografii

Konvexní obálky

Bc. Petra Pasovská
Bc. David Zahradník

Obsah

1 Zadání

Níže uvedené zadání je kopie ze stránek předmětu.

Vstup: množina $P = \{p_1, ..., p_n\}$, $p_i = [x, y_i]$.

Výstup: $\mathcal{H}(P)$.

Nad množinou P implementujete následující algoritmy pro konstrukci $\mathcal{H}(P)$:

- Jarvis Scan,
- Quick Hull,
- Swep Line.

Vstupní množiny bodů včetně vygenerovaných konvexních obálek vhodně vizualizujte. Pro množiny $n \in < 1000, 1000000 >$ vytvořte grafy ilustrující doby běhu algoritmů pro zvolená n. Měření proveďte pro různé typy vstupních množin (náhodná množina, rastr, body na kružnici) opakovaně (10x) a různá n (nejméně 10 množin) s uvedením rozptylu. Naměřené údaje uspořádejte do přehledných tabulek.

Zamyslete se nad problematikou možných singularit pro různé typy vstupních množin a možnými optimalizacemi. Zhodnoťte dosažené výsledky. Rozhodněte, která z těchto metod je s ohledem na časovou složitost a typ vstupní množiny P nejvhodnější.

Hodnocení:

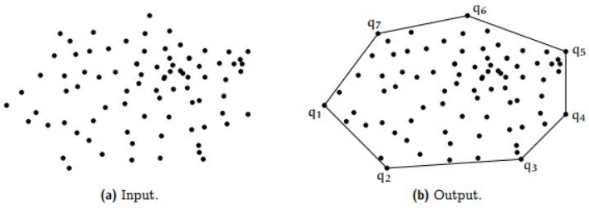
Krok	**Hodnocení**
Konstrukce konvexních obálek metodami Jarvis Scan, Quick Hull, Seep Line.	15b
Konstrukce konvexní obálky metodou Graham Scan	*+5b*
Konstrukce striktně konvexních obálek pro všechny uvedené algoritmy.	*+5b*
Ošetření singulárního případu u Jarvis Scan: existence kolineárních bodů v datasetu.	*+2b*
Konstrukce Minimum Area Enclosing box některou z metod (hlavní směry budov).	*+5b*
Algoritmus pro automatické generování konvexních/nekonvexních množin bodů různých tvarů (kruh, elipsa, čtverec, star-shaped, popř. další).	*+4b*
Max celkem:	**36b**

1.1 Údaje o bonusových úlohách

2 Popis a rozbor problému

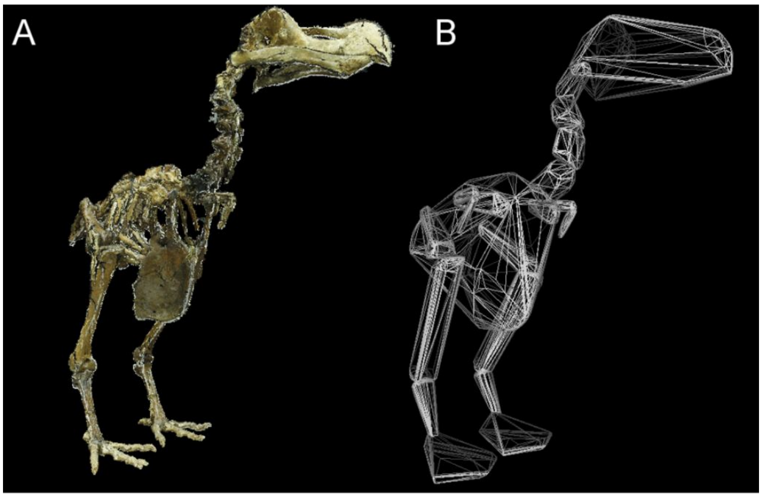
Hlavním cílem této úlohy je tvorba aplikace, která pro vygenerované množství bodů vytvoří konvexní obálku za pomoci různých algoritmů. Pro jednotlivé metody byla zapisována i doba trvání algoritmu. Výsledné časy jsou v závěru následně porovnány.

Lze říci, že konvexní obálka množiny M je nejmenší konvexní množina, která množinu M obsahuje. V současné době mají konvexní obálky, v některých literaturách označovány jako konvexní obaly, mnoho využití. Často se využívají jako první odhad tvaru nějakého prostorového jevu, např. detekce kolizí, detekce natočení budov a jejich tvaru v kartografii, analýza shluků atd. [Zdroj: 1]



Obrázek 1: Ukázka vstupních bodů, kolem nichž je vytvořena konvexní obálka. [zdroj: 2]

Konvexní obálky využívá celá řada vědních oborů. Zajímavé bylo využití konvexních obálek v paleontologii, kde za pomoci konvexních obálek jsou vědci schopni určit přibližně tvar těla vyhynulých živočichů, jejichž kosti byly nalezeny.



Obrázek 2: Využití konvexních obálek v paleontologii [zdroj: 3]

3 Popis použitých algoritmů

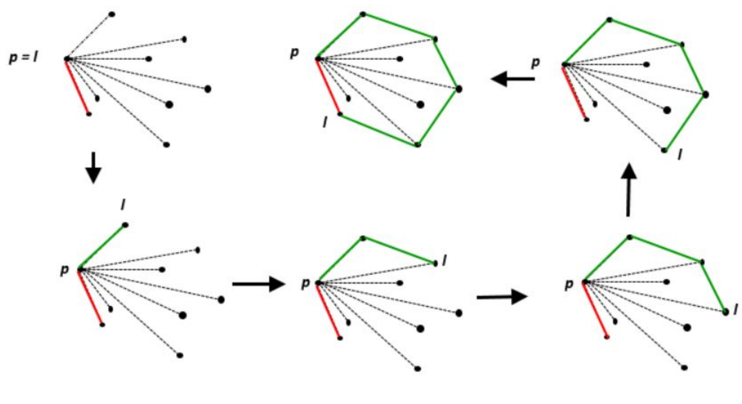
Existuje několik způsobů jak vytvořit konvexní obálku. V této úloze byly použity pro tovrbu 3 metody. (4 když se povede bonus????)

3.1 Jarvis Scan

Tato metoda bývá přirovnávána ke způsobu balení dárků (alternativní název Gift Wrapping Algorithm). Předpokladem pro algoritmus Jarvis Scan je, že 3 body nesmí ležet na jedné přímce. Metoda je poměrně snadná pro zápis, velkou nevýhodou je však časová náročnost O(n^2), které lze dosáhnout, pokud body z množiny S leží na kružnici. Běžný čas výpočtu bývá O(n*h), kde n je počet vstupních bodů a h je počet bodů, které tvoří obálku. [zdroj: 1]

Metoda je pojmenována po R. A. Jarvisu, který ji publikoval v roce 1973.

Abychom byli schopni algoritmus sestavit, je potřeba nalézt pivot, označme jej q. Nalezení pivota má časovou náročnost O(n). Pivota nalezneme jako bod s minimální hodnotou souřadnice Y. Následně porovnáváme úhel, který svírá pivot a bod následující a předcházející pivotu, dokud nenalezneme maximální úhel. Když takovýto úhel nalezeneme, je přidán mezi body konvexní obálky. V algoritmu dojde k přeindexování bodů a jsou porovnávány následující body, dokud nově vložený bod není pivot.



Obrázek 3: Princip Jarvis Scan algoritmu [zdroj: 4]

3.1.1 Problematické situace

K chybě v algoritmu může dojít v případě, že tři body budou kolineární. Tedy v případě, že se budou tři body nacházet na jedné přímce. (Jak bychom toto vyřešili hmm??)

3.1.2 Implementace metody

1. Nalezení pivota q: $q = min(y_i)$
2. Přidej bod q do konvexní obálky: $q \rightarrow H$

3. Inicializuj: $p_j = q; p_{j+1} = p_{j-1}$
4. Opakuj, dokud: $p_{j+1} \neq q$

   Nalezni p_{j+1}: $p_{j+1} = argmax_{\forall p_i \in P} \angle(p_{j-1}, p_j, p_i)$

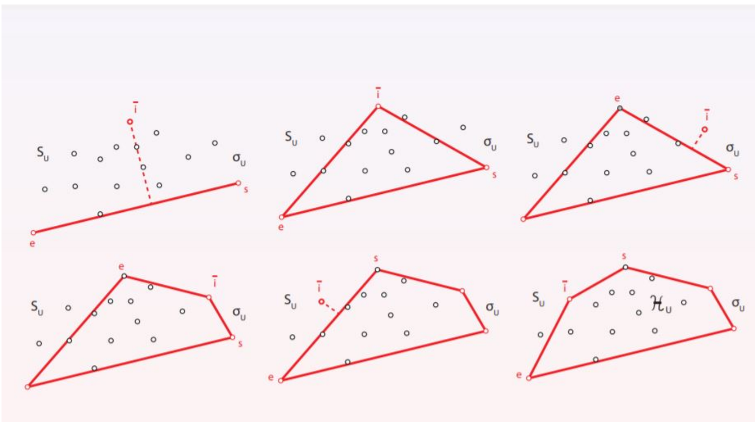
   Přidej p_{j+1}: $p_{j+1} \rightarrow H$

   Přeindexování bodů: $p_{j-1} = p_j; p_j = p_{j+1}$

3.2 Quick Hull

Metoda Quick Hull slouží k vytvoření konvexní obálky nad konečným počtem bodů. Využívá techniku "Rozděl a panuj", označovanou anglicky "Divide and Conquer". V této metodě lze najít analogii s QuickSortem, odkud také pochází označení algoritmu. Jedná se o poměrně rychlý algoritmus, časová náročnost je O(n*log(n)), v nejhorším případě je však časová náročnost kvadratická O(n^2).

Pro výpočet metodou Quick Hull je nejprve zapotřebí nalézt extrémní body, v aplikaci byly souřadnice setříděny podle x-ové souřadnice. Bod s nejnižší a nejvyšší hodnotou x-ové souřadnice je vložen do množiny, do které uchováváme body konvexní obálky. Těmito body je vedena pomyslná přímka, která množinu bodů rozdělí na dvě množiny - horní a dolní. V každé polorovině nalezneme nejvzdálenější bod od přímky, tento bod přidáme do množiny bodů patřících do konvexní obálky a vytvoříme přímky tohoto bodu a krajních bodů předešlé přímky. Následně pokračujeme analogicky a nad každou nově vzniklou přímkou nalezneme nejvzdálenější bod.



Obrázek 4: Princip Quick Hull algoritmu [zdroj: 5]

3.2.1 Implementace globální metody

1. Vytvoření množiny konvexní obálky, horní a dolní množiny: $H = 0; S_U = 0; S_L = 0$
2. Nalezení extrémních hodnot: $q_1 = min_{\forall p_i \in S}(x_i); q_3 = max_{\forall p_i \in S}(x_i)$
3. Přidání extrémních bodů do horní a dolní množiny: $S_U \leftarrow q_1; S_U \leftarrow q_3; S_L \leftarrow q_1; S_L \leftarrow q_3$

4. Pro všechny body množiny: $\forall p_i \in S$

   Rozhodnutí, zda bod patří do horní množiny: $if(p_i \in \sigma_l(q_1, q_3))S_U \leftarrow p_i$

   V opačném případě: $S_L \leftarrow p_i$

5. Přidání krajního bodu do konvexní obálky: $H \leftarrow q_3$

6. Nalezení nejvzdálenějšího bodu c v horní části od přímky, přidání do množiny konvexní obálky a opakování vůči nově vzniklé přímce.

7. Přidání krajního bodu do konvexní obálky: $H \leftarrow q_1$

8. Opakování hledání nejvzdálenějšího bodu v dolní části.

Tady nevím, jestli nerozepsat i implementaci lokální metody??? Možná by to pak bylo pochopitelnější

4 Vstupní data

5 Výstupní data

6 Aplikace

7 Dokumentace

8 Závěr

9 Reference

1. MARTÍNEK, Petr. Konvexní obálka rozsáhlé množiny bodů v Eđ [online][cit. 31.10.2018]. Dostupné z: http://graphics.zcu.cz/files/86_BP_2010_Martinek_Petr.pdf

2. Convex Hulls: Explained. [online][cit. 31.10.2018] Dostupné z: https://medium.com/@harshitsikchi/convex-hulls-explained-baab662c4e94

3. Convex-hull mass estimates of the dodo (Raphus cucullatus). [online][cit. 31.10.2018] Dostupné z: https://www.semanticscholar.org/paper/Convex-hull-mass-estimates-of-the-dodo-(Raphus-of-a-Brassey-OḾahoney/12e07d3b712561cad16501ac8096120e14901eb8

4. GeeksforGeeks: Convex Hull - Set 1 (Jarvis's Algorithm or Wrapping. [online][cit. 5.11.2018] Dostupné z: https://www.geeksforgeeks.org/convex-hull-set-1-jarviss-algorithm-or-wrapping/

5. BAYER, Tomáš. Geometrické vyhledávání [online][cit. 5.11.2018]. Dostupné z: https://web.natur.cuni.cz/ bayertom/images/courses/Adk/adk4.pdf