155ADKG: Konvexní obálky a jejich konstrukce

Datum odevzdání: 26.11.2017

Petra Millarová, Oleksiy Maybrodskyy

Contents

1 Zadání

Následuje kopie oficiálního zadání úlohy. Autoři z nepovinných bodů zadání úspěšně implementovali algoritmus pro ošetření singulárního případu u Jarvis Scan: existence kolineárních bodů v data, rozpracovana kontrukce konvexní obálky metodou Graham Scan.

Úloha č. 2: Konvexní obálky a jejich konstrukce

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.
- Incremental Construction.

Vstupní množiny bodů včetně vygenerovaných konvexních obálek vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku QT. Dynamické datové struktury implementujte s využitím knihovny STL.

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, clustrovaná data) opakovaně (10x) a různá n (celkem 10) 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, Incremental Construction.	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.	*+5b*
Algoritmus pro automatické generování konvexních/nekonvexních množin bodů různých tvarů (kruh, elipsa, čtverec, hvězda, popř. další).	*+4b*
Max celkem:	**36b**

Čas zpracování: 2 týdny.

2 Popis a rozbor problému

Tato úloha se věnuje řešení praktického problému konvexní obálky pro náhodné množiny bodů, pravidelné množiny bodů. Jako implementaci si lze zjednodušeně představit vytvarování digitální mapy.

3 Popisy algoritmů

V dané úloze jsou použity následující algoritmy, avšak existují i další možnosti, jak polohu bodu určit.

3.1 Jarvis Scan

Jedna se o dost pomalý algoritmus oprotí ostatním algoritmům zabývájicím se obdobním problémy. Algoritmus funguje na principu gip wrapping, česky balení dárku.
Nechť v kartezské soustavě souřadnic existuje množina bodů. Setříděním souřadnic dle osy $\boldsymbol{Y}$, je možné

najít počáteční bod $\boldsymbol{q}$ (dále jen pivot), který je dan obrázek bodu s nejmenší hodnotou na ose $\boldsymbol{Y}$. Dále předpokladáme, že v množině uvedených bodů neexistují 3 kolineární body. Pak je možné aplikovat níže uvedený algoritmus:

1. Nalezení pivota $q = \min(y_i)$,
2. Vklad q do množiny H.
3. p_j = q, $p_i = p_{j-1}$.
4. Načítání bodu p_{j-1}
5. $p_i \neq$ q:
6. Cyklus pro body p_{j-1}, p_j:
7. Dokud p_i je takové, že $\theta = \min(\theta_i)$;
8. Vklad do množiny H.

3.2 Quick Hull

Algoritmus typu QuikSort. Konvexní obálka tvořena z horní a dolní částí.
Horní část obsahuje body nad spojnicí bodů a dolní část je pod spojnicí q_1 a q_3.
Řešení pro každou konvexní obálku se provádí zvlašť a následně se slučuje. Hledáme nejvzdalenější bod pro každou konvexní obálku ležící vpravo od této strany. Nově vzniklý bod se stává bodem obálky. Nově vzniklá strana se rozděluje na dvě nové strany, princip rozděluj a panuj.

3.3 Incremental Construcion

Inkrementální algoritmus je velmi rychlý algoritmus pro výpočet konvexní obalky. Princip se základa na postupným testováním bodů, respektive jejich předaváním do konvexní obalky, jejiž tvar je modefikovan.

3.4 Graham Scan

Grahamuv Scan je algoritmus určený pro výpočet konvexní obálky ve dvou rozměrném prostoru.
Algoritmus nejprve vyhledává pivot p , jehož souřadnice y je minimální. Z něj se proloži polární orientace na ostatní body. Body jsou následně seřazeny dle velikostí těchto orientaci, respektivě úhlů. Pivot a všechny body co leží vlevo, jsou postupně ukládany do konvexní obálky.

3.5 Striktně konvexní obálka

Účelem algoritmu je detikovat body na přímkach, respektivě, aby přimka jako taková, byla tvořena jen 2 body, tedy počátkem a koncem.
V algoritmu jsou postupně procházeny všechny trojíce bodů. Pokud se prokazalo, že 3 body leží na společní přímce, tak prostřední bod je detikovan.

4 Vstupní data

Aplikace má 2 mody. První mod znázorňuje výsledky výpočtu obrázem situace.Mezi vstupními daty patří jediná hodnota, a to koknretně uživatelem zadaná hodnota požadovaného počtu bodů. Generovát lze na základě randomného, respektve náhodného rozložení, nebo na základě gridu, respektive pravidelné síti bodů. V rámci třetí možností je třeba ještě uvést náhodného generování shluku bodů. Druhý mod, respektive Graph

mode vytváří graf, který znázorňuje generování v konkretních metodach v intervalu od 1000 do 1000000 a úkláda výsledky grafů do formatu .PDF.

5 Výstupní data

Výstup je vizualizací řešeného problému v grafickém okně. Taky v grafickém okně je slovně napsana rychlost výpočtu algoritmu.
Praktickým výstupem je graf znázorňující rychlot výpočtu algoritmu v podobě vykreslování grafu. Uživátel

si sam zvolí umistění. Víc další kapitoly

6 Ukázky aplikace

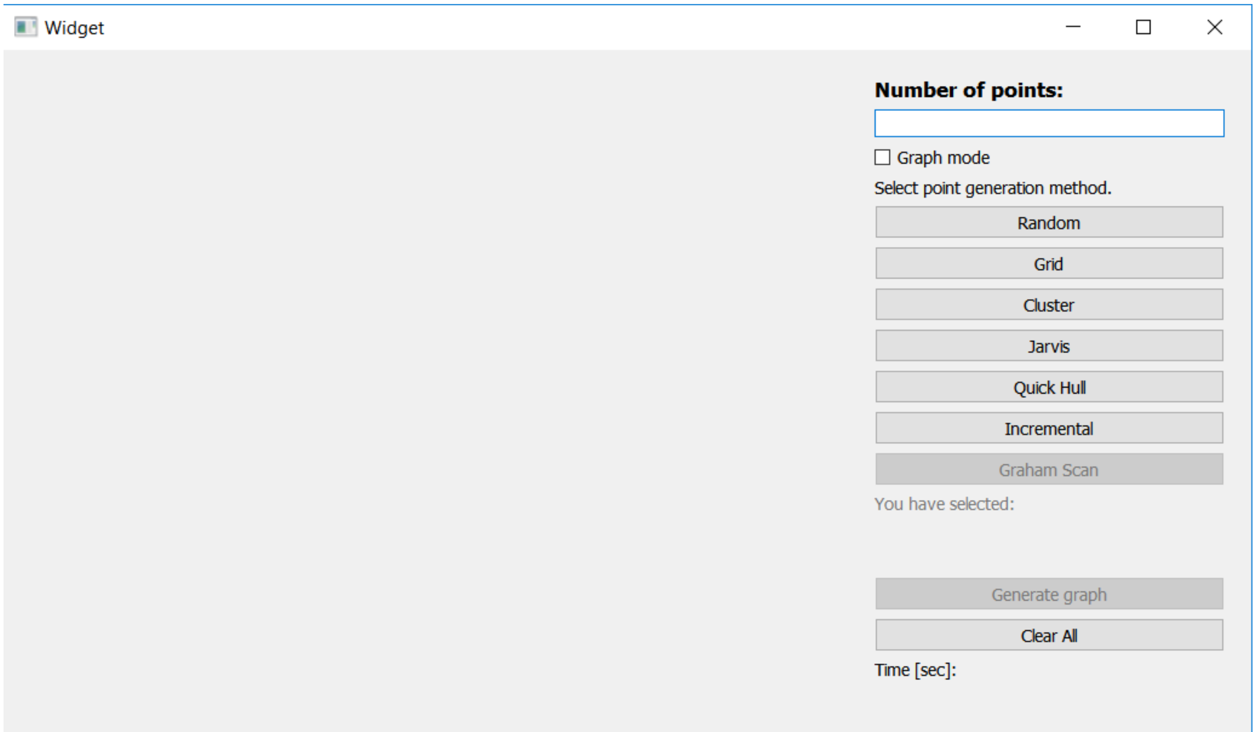

Figure 1: Aplikace po spuštění

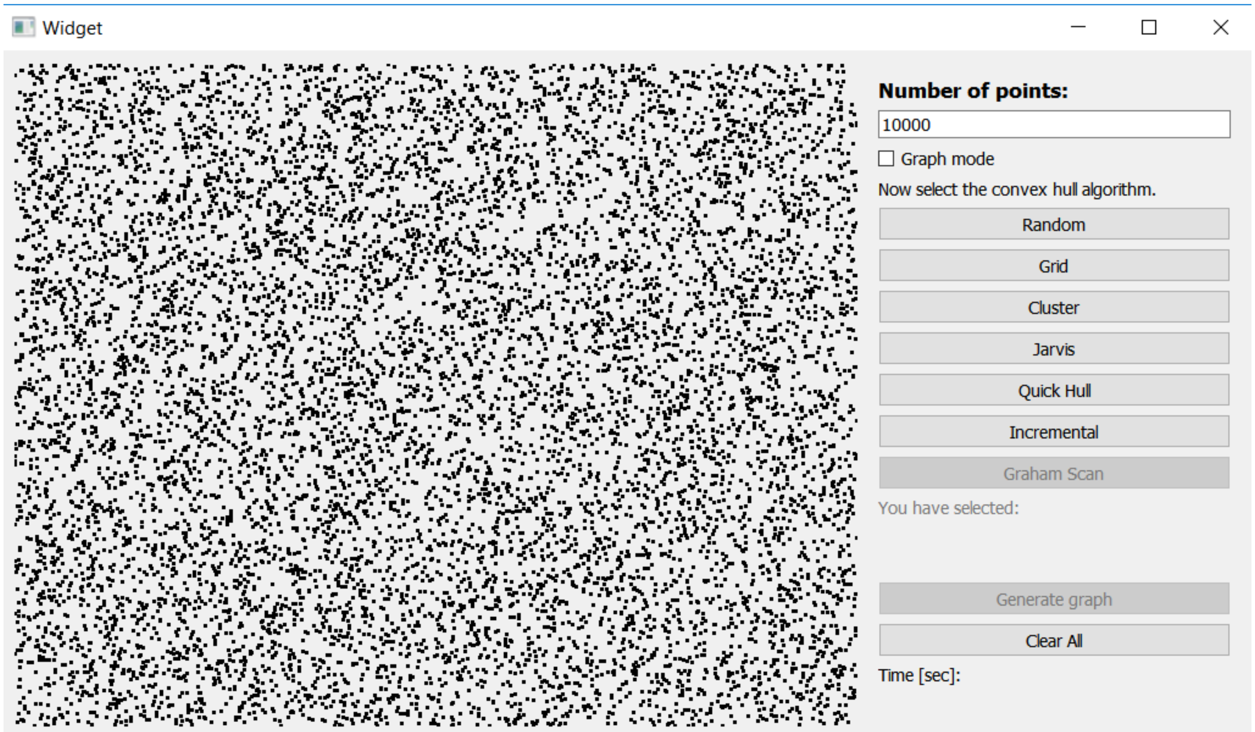

Figure 2: Aplikace po spuštění RANDOM

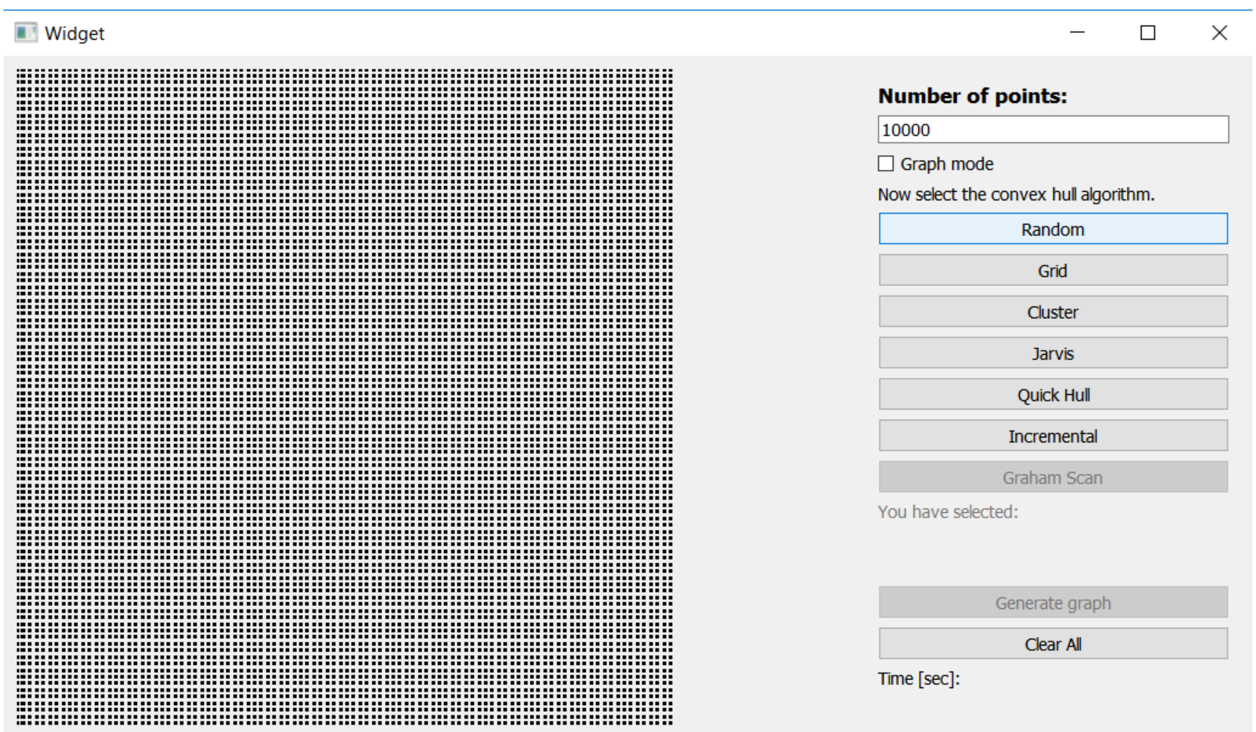

Figure 3: Aplikace po spuštění GRID

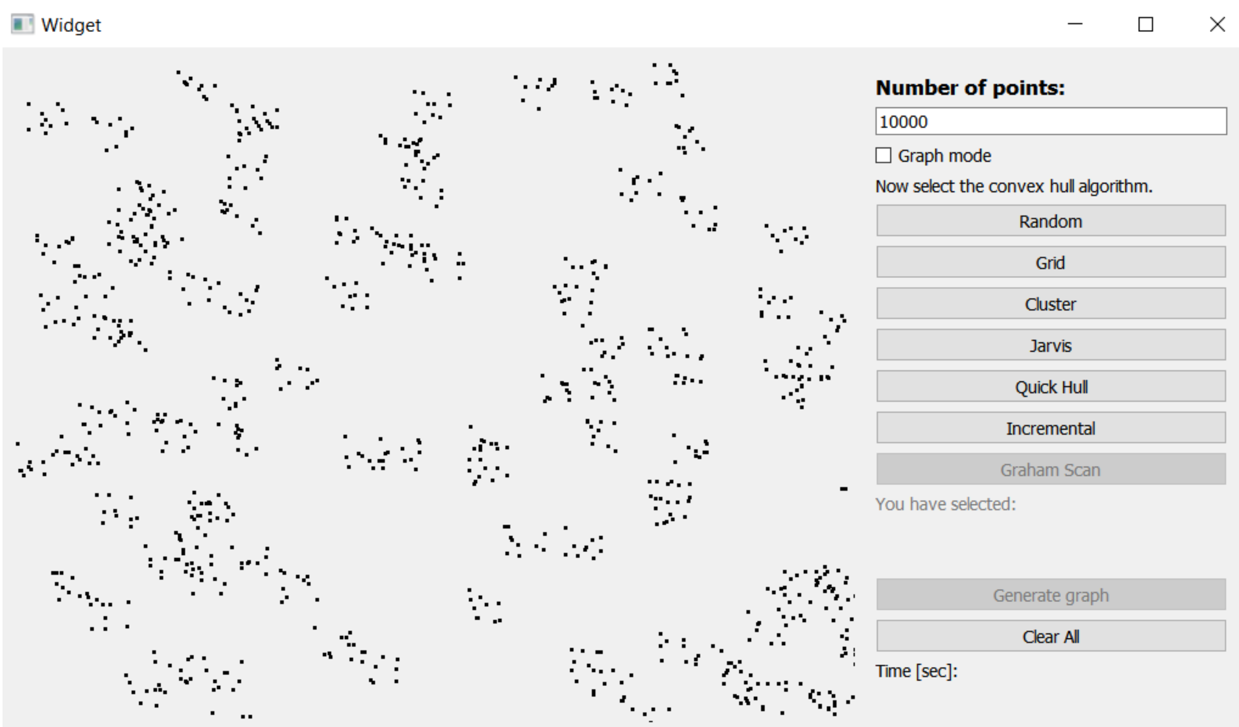

Figure 4: Aplikace po spuštění Cluster

6.1 RANDOM

Jarvis Scan

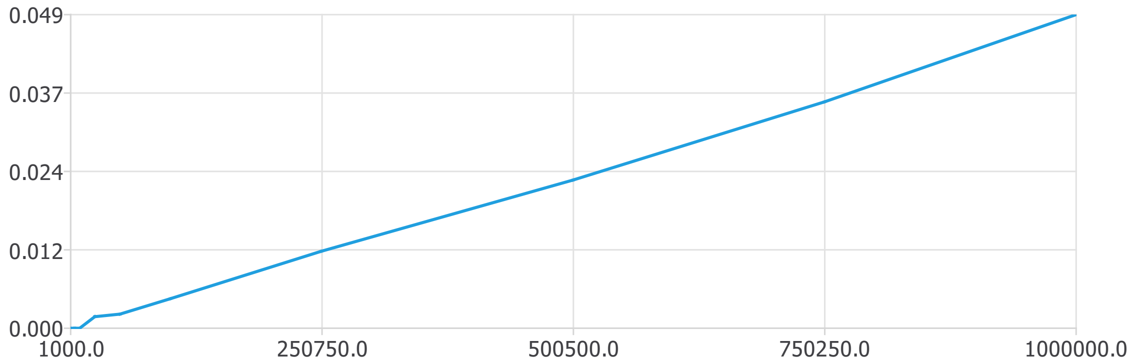

Figure 5: generování jednou

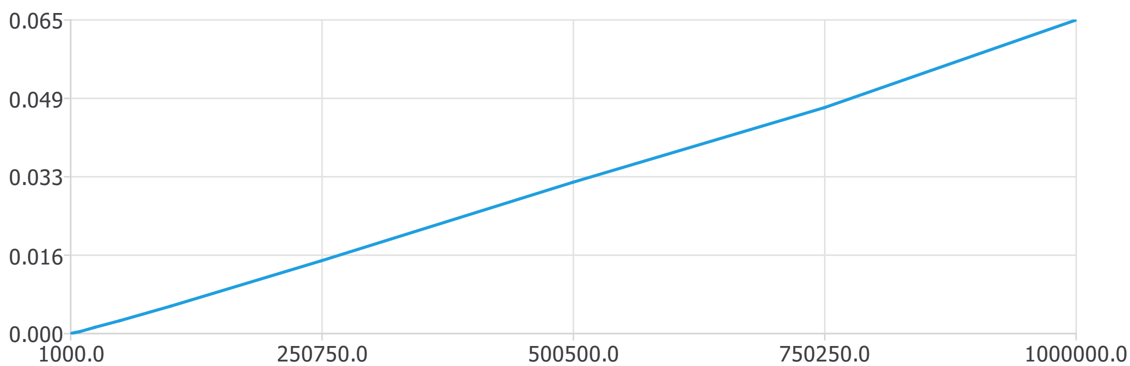

Figure 6: průměr z 10 generování

Quick Hull

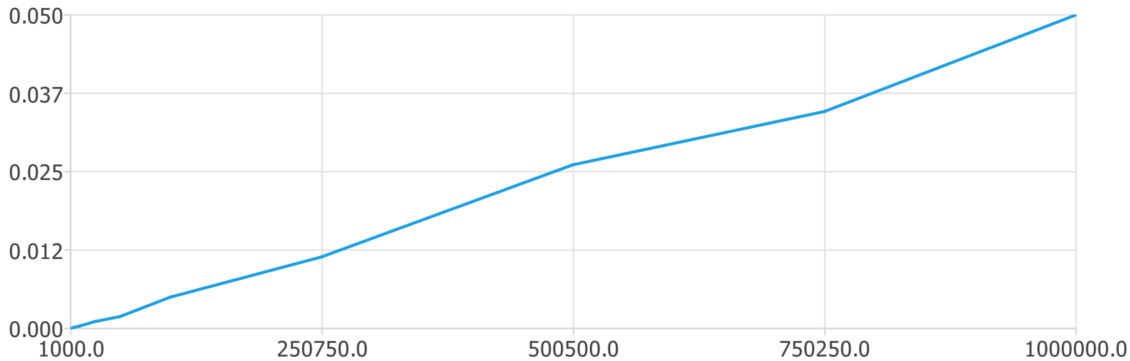

Figure 7: generování

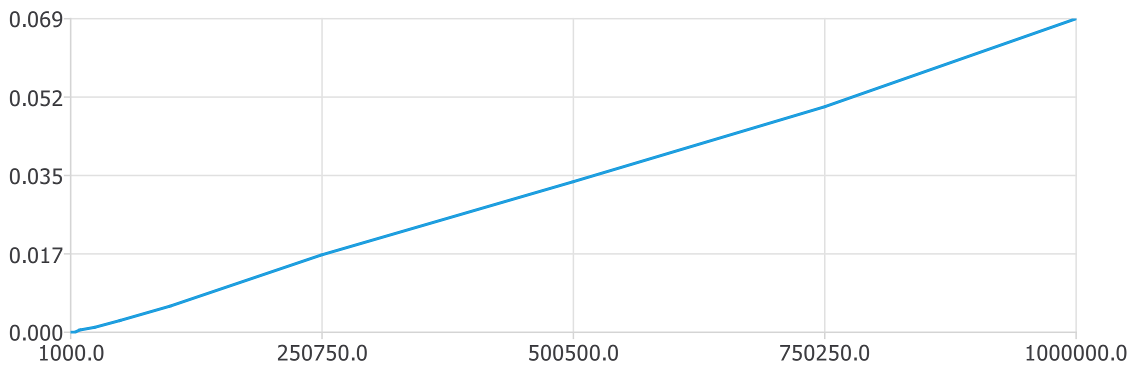

Figure 8: průměr z 10 generování

Incremental construction

.

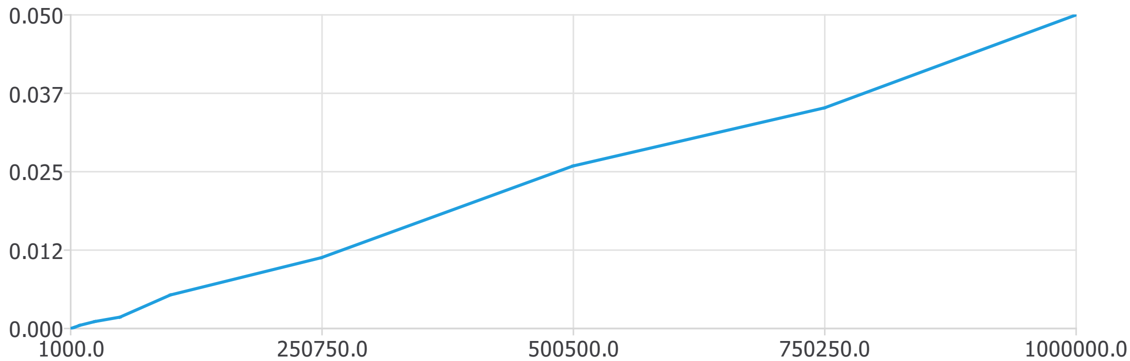

Figure 9: generování

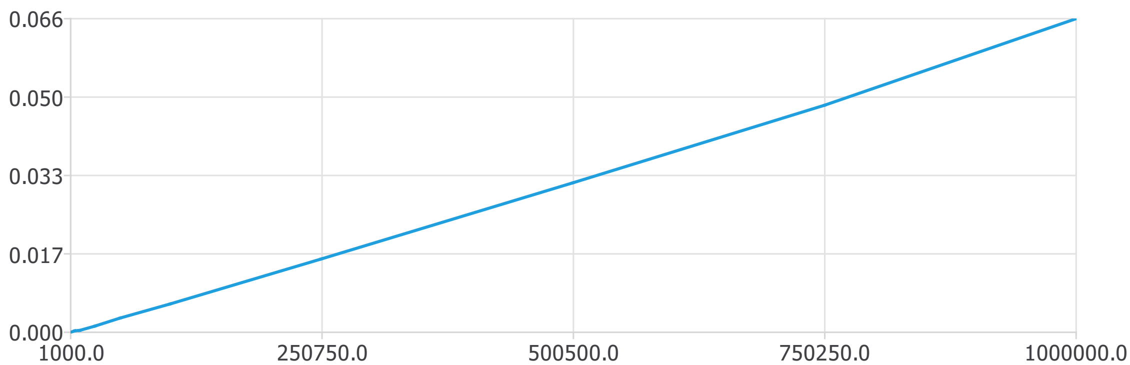

Figure 10: průměr z 10 generování

Graham Scan

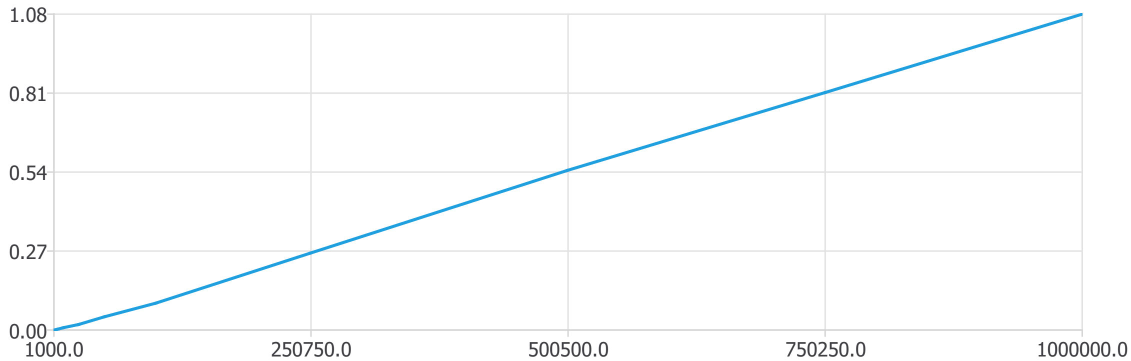

Figure 11: průměr z 10 generování

6.2 GRID

Jarvis Scan

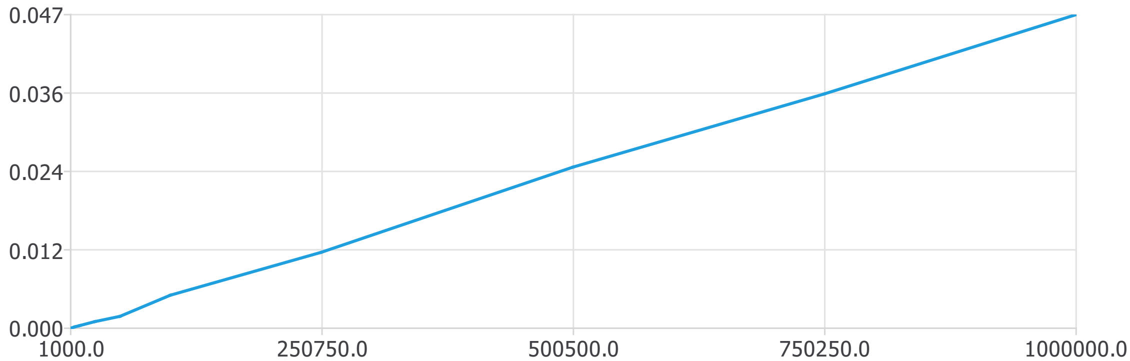

Figure 12: generování

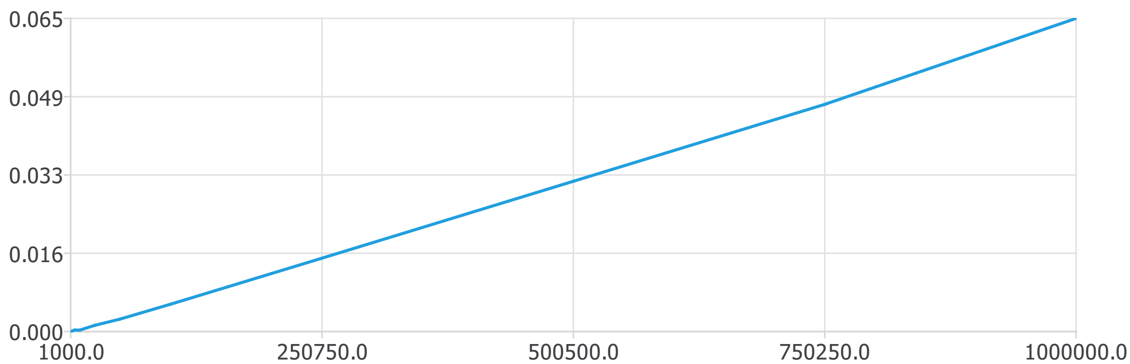

Figure 13: průměr z 10 generování

Quick Hull

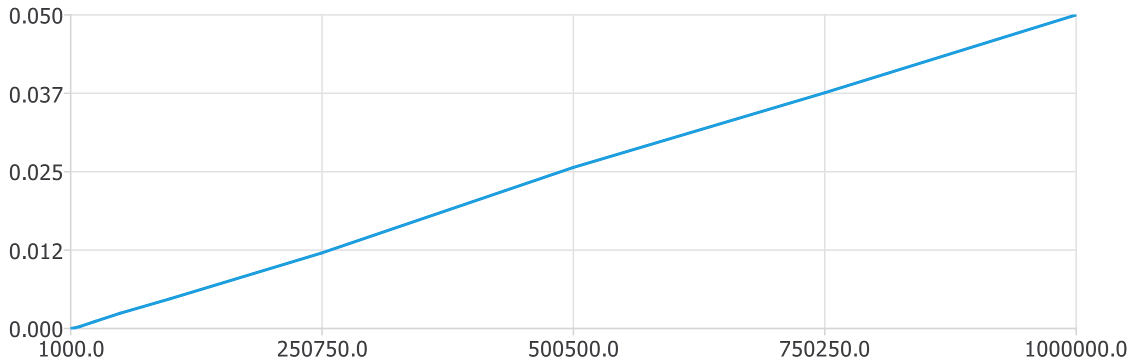

Figure 14: generování

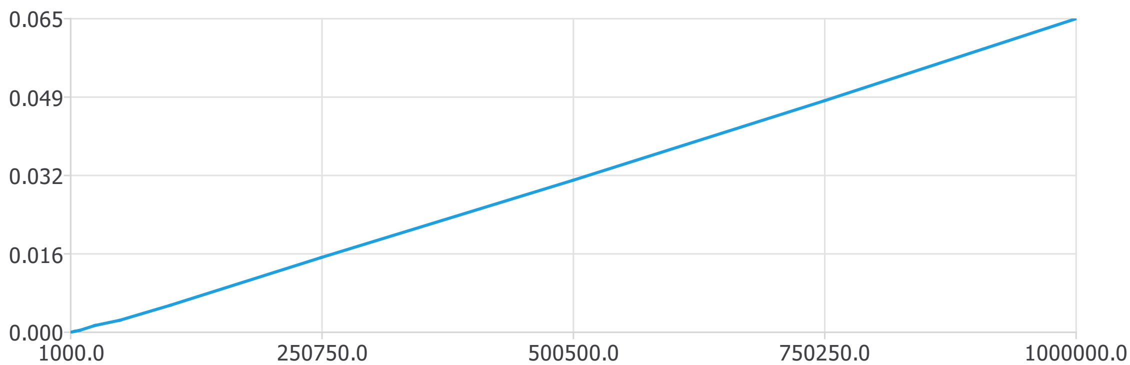

Figure 15: průměr z 10 generování

Incremental construction

.

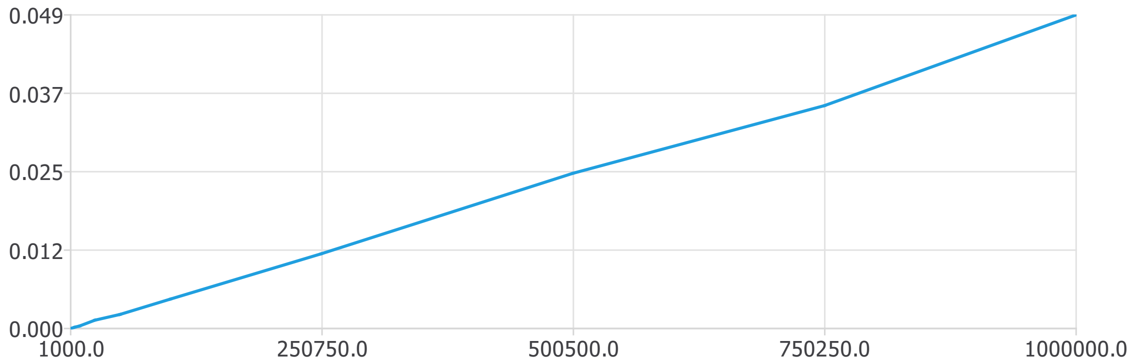

Figure 16: generování

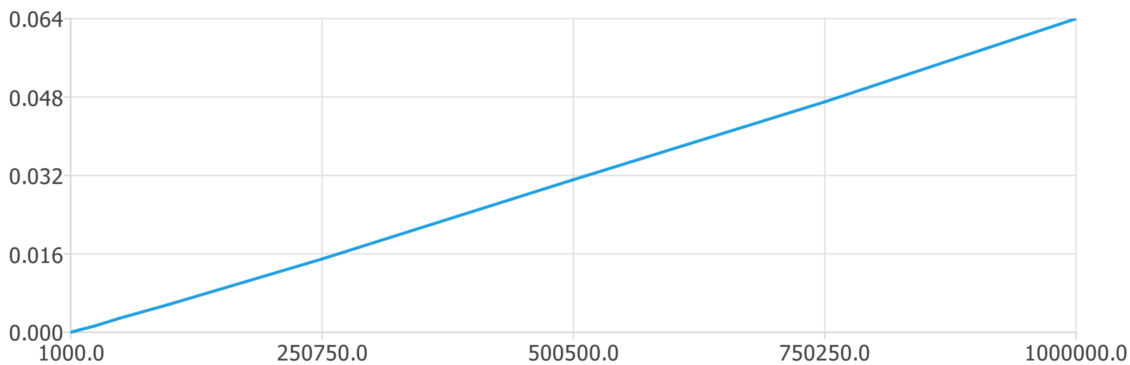

Figure 17: průměr z 10 generování

Graham Scan

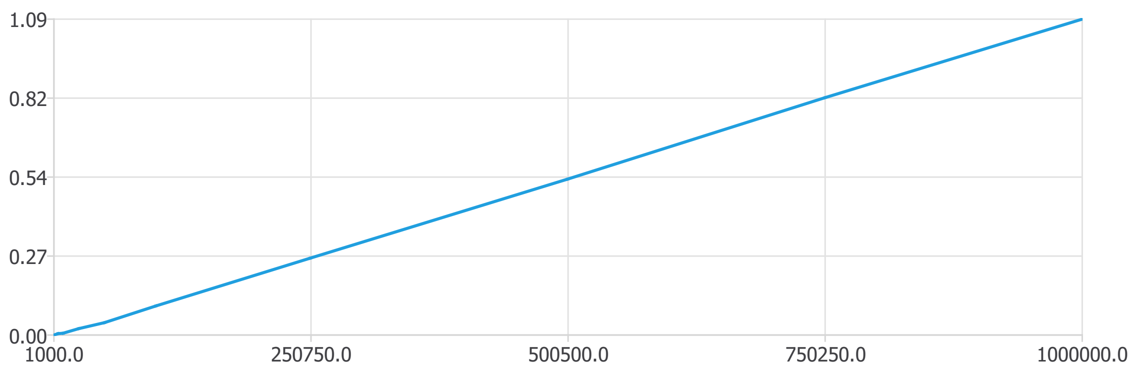

Figure 18: průměr z 10 generování

6.3 CLUSTER

Jarvis Scan

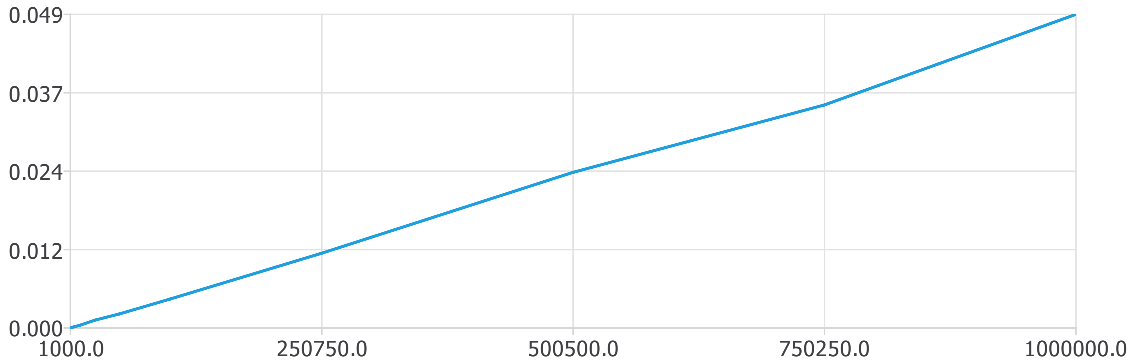

Figure 19: generování

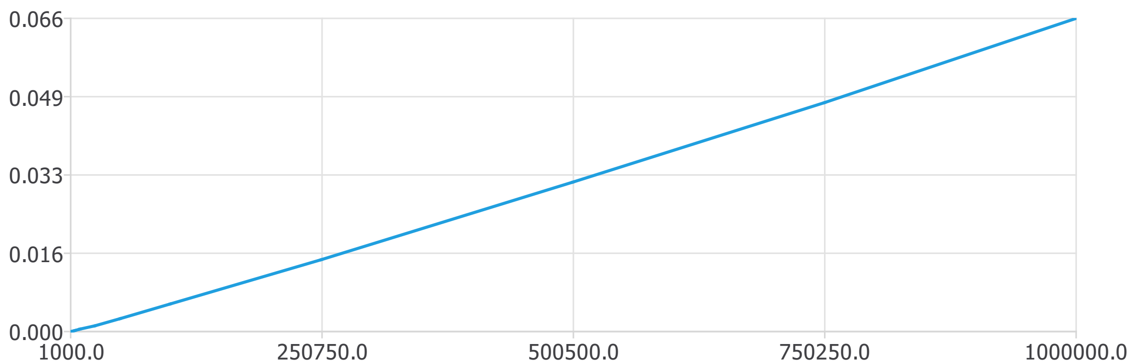

Figure 20: průměr z 10 generování

Quick Hull

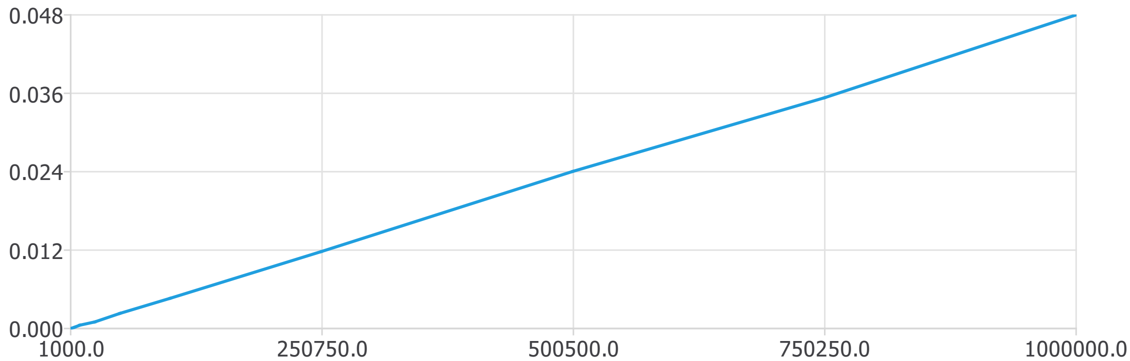

Figure 21: generování

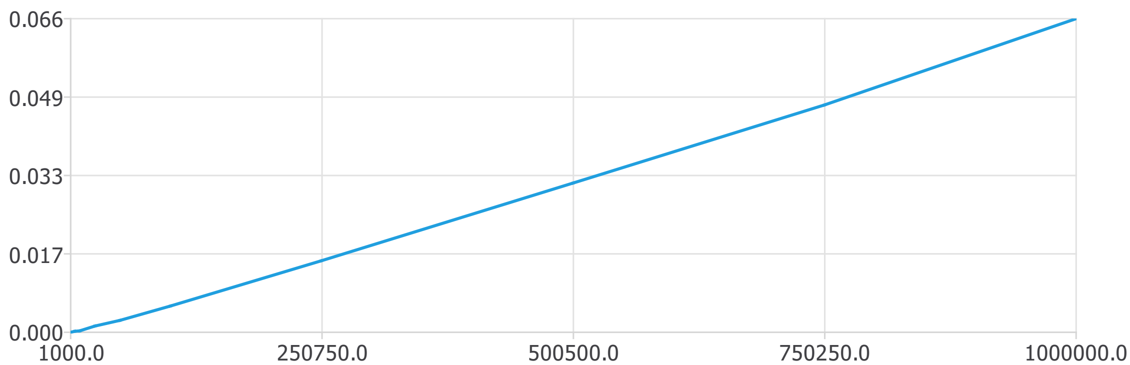

Figure 22: průměr z 10 generování

Incremental construction

.

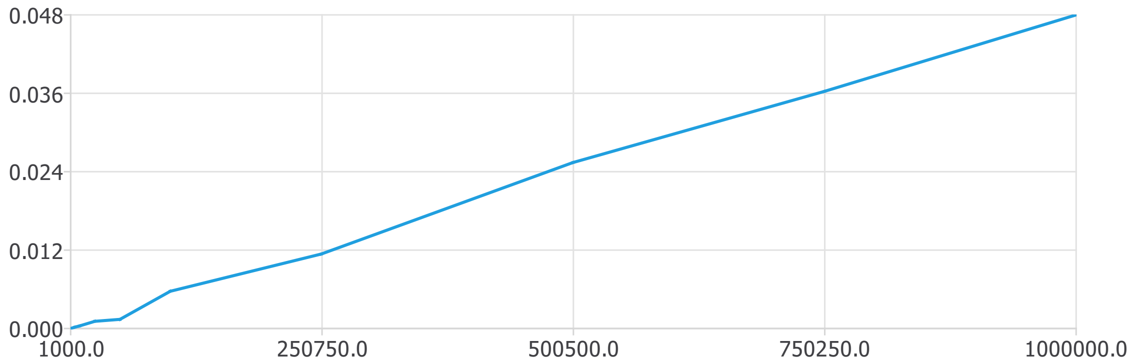

Figure 23: generování

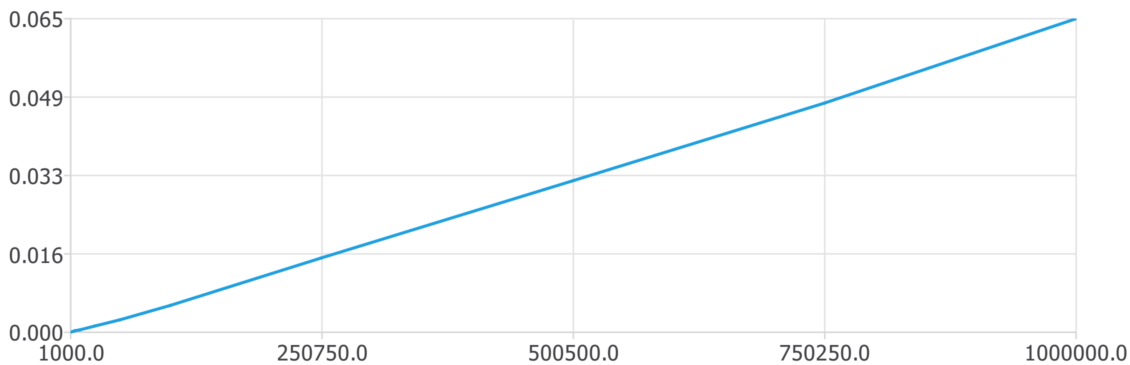

Figure 24: průměr z 10 generování

Graham Scan

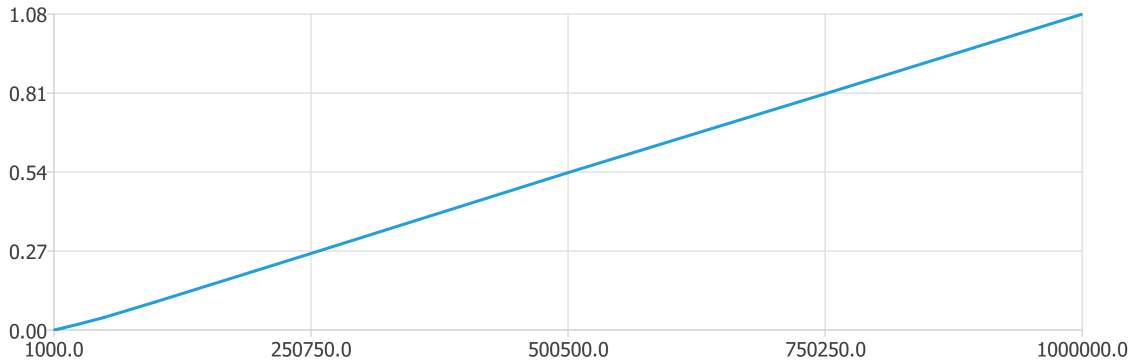

Figure 25: průměr z 10 generování

7 Závěr

Autoři splnili většinu bodů zadání a vznikl program, který generuje soubor bodů, jejich rozmíštění a počita rychlost výpočtu algortimu. V předchozí kapitole jsou zobrazeny grafy, které znázorňují pruběhy výpočtu algoritmů. Test byl proveden několikrát. Jak je vidět z grafů, nejvíce nestabilini je Jarvis Scan. Dochází k velkému kolisání v počátečných aj dokonce u středních hodnot. O něco lépe je na tom Quick Hull, a však aj u něho dochází ke značným výchylům. Obecně dalo by se řici, že Incremental construction je nejvíc stabilní algoritmus ze všech výše uvedených. Každopádně k výchylům na intervalu hodnot kolem několika

tisíc dochází pravidelně a u všech algoritmů. S rostoucím počtem n klesa zároveň výchylka mezi jednotlivými testy. Zároveň s rostoucím počtem n klesa rozdíl mezi jednotilivými metodami. Z výsledku lze usoudit, že

pro generování menšího objemu dat je lepší využit Incremental construction, případně Quick Hall, naopak pro velký objem dat co převýšuje velikosti desítek tisíc je možné využivát jaký koliv algoritmus. Graham Scan se prokazuje jakož to nevhodný a zpomalený algoritmus. Z bonusových úloh bylo naprogramováno:
Graham Scan
Konstrukce striktně konvexních obálek pro všechny algoritmy
Ošetření singulárního připadu u Jarvis Scan

7.1 Náměty na vylepšení

Aplikace, ač funkční a splňující daný účel, má spoustu nedostatků, které by bylo dobré v budoucnu odstranit. Autoři zde uvádí pár těch nejzjevnějších.
V algoritmu Graham Scan pro Grid siť dochází k tomu, že občas jeden krajný dolný bod není zahrnut do

vzníku nové obálky. Bohužel chybu nešlo přimo detekovát.
Souřadnicové osy:

Vykreslovací okno má v Qt, stejně jako ve většině podobných nástrojů, počátek souřadnic v levém horním rohu, kladnou osu x vpravo a kladnou osu y směrem dolů. Tento model se však neshoduje ani s geodetickými souřadnicemi používanými na našem území (kladná y doleva, kladná x dolů), ani s klasickým označením os (kladná x doprava, kladná y nahoru). Proto se body v současné verzi zobrazují jinak, než by možná uživatel očekával. Vhodným řešením by byla transformace.