České vysoké učení v Praze

Fakulta stavební

155ADKG - Algoritmy digitální kartografie

Úloha 3: Digitální model terénu

Radek Novotný

Zadání:

Úloha č. 3: Digitální model terénu

Vstup: množina $P = \{p_1, ..., p_n\}$, $p_i = \{x_i, y_i, z_i\}$.

Výstup: polyedrický DMT nad množinou P představovaný vrstevnicemi doplněný vizualizací sklonu trojúhelníků a jejich expozicí.

Metodou inkrementální konstrukce vytvořte nad množinou P vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů) popř. navrhněte algoritmus pro generování syntetických vstupních dat představujících významné terénní tvary (kupa, údolí, spočinek, hřbet, ...).

Vstupní množiny bodů včetně níže uvedených výstupů vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku QT. Dynamické datové struktury implementujte s využitím STL.

Nad takto vzniklou triangulací vygenerujte polyedrický digitální model terénu. Dále proveďte tyto analýzy:

- S využitím lineární interpolace vygenerujte vrstevnice se *zadaným krokem* a v *zadaném intervalu*, proveďte jejich vizualizaci s rozlišením zvýrazněných vrstevnic.
- Analyzujte sklon digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich sklonu.
- Analyzujte expozici digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich expozici ke světové straně.

Zhodnoťte výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabinami algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích (různé terénní tvary) nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

Zhodnocení činnosti algoritmu včetně ukázek proveďte alespoň na tři strany formátu A4.

Hodnocení:

Krok	**Hodnocení**
Delaunay triangulace, polyedrický model terénu.	10b
Konstrukce vrstevnic, analýza sklonu a expozice.	10b
Triangulace nekonvexní oblasti zadané polygonem.	*+5b*
Výběr barevných stupnic při vizualizaci sklonu a expozice.	*+3b*
Automatický popis vrstevnic.	*+3b*
Automatický popis vrstevnic respektující kartografické zásady (orientace, vhodné rozložení).	*+10b*
Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet, ...).	*+10b*
3D vizualizace terénu s využitím promítání.	*+10b*
Barevná hypsometrie.	*+5b*
Max celkem:	**65b**

Čas zpracování: 3 týdny

Bonusové úlohy:

V rámci bonusových úloh byla vytvořena možnost vygenerovat body tvořící kupu.

Popis problému a algoritmů:

Nad množinou bodů, jež mají 3 souřadnice x, y, z, je potřeba vytvořit pomocí Delaunay tringulace síť trojúhelníků a na této síti vytvořit lineární interpolací síť vrstevnic. Dále je potřeba vizualizovat sklon a expozici jednotlivých trojúhelníků.

Vytvořená trojúhelníková síť musí splňovat následující podmínky:

- Libovolné dva trojúhelníky mají nejvýše společnou hranu.
- Uvnitř žádného trojúhelníku neleží žádný další bod, který by měl být v síti.

Triangulace se promítá do mnoha oblastí dnešní doby. Od námi nejbližší kartografie a GIS – tvorba digitálních modelů terénu nebo zpracování dat z dálkového průzkumu např. laserového mračna bodů. Až po detekci otisků prstů v biometrii nebo vizualizace prostorových dat v počítačové grafice.

Obecně triangulační algoritmy patří mezi jedny z nejvíce teoreticky rozpracovaných postupů.

<u>Delaunay Triangulace</u>

Delaynay triangulace je nejčastěji používanou triangulací v rámci geografických informačních systémů (GIS), zejména při tvorbě digitálního modelu terénu (DMT), ovšem lze provádět nejen v prostoru, ale i v rovině.

Delaunay triangulace má následující vlastnosti:

- Uvnitř kružnice opsané libovolnému trojúhelníku triangulace neleží žádný jiný bod
- Maximalizuje minimální úhel, avšak neminimalizuje maximální úhel v trojúhelníku.
- Vůči kritériu minimálního úhlu je lokálně i globálně optimální
- Triangulace je jednoznačná, pokud čtyři body neleží na kružnici

Triangulace byla vytvořena inkrementální konstrukcí, body jsou do sítě přidávány postupně, přičemž musí splňovat tyto podmínky: ležet v levé polorovině od orientované strany, minimální poloměr opsané kružnice. Pokud žádný bod nevyhovuje je orientace obrácena a bod volen znovu. Po vybrání bodu jsou hrany k němu vedoucí od orientované hrany uloženy.

Pro hledání těchto hran je využita struktura AEL – Active Edges List, v níž máme hrany, pro něž je potřeba nalézt třetí bod. Algoritmus tedy končí po vyprázdnění tohoto listu.

Algoritmus 1: Delaunay Triangulation Local(P)

```
1: Vytvoř pomocnou triangulaci T(P).
2: legal=false
3: while T(P) !legal
4:    legal=true;
5:    Opakuj pro ∀ei ∈ T(P)
6:    Vezmi hranu ei ∈ T(P)
7:    Nalezni trojúhelníky t1, t2 incidující s ei .
8:    if (t1 ∪ t2) konvexní a nelegální
9:          Legalize (t1, t2).
10:         legal=false;
```

Algoritmus 2: Delaunay Triangulation Incremental (P, AEL, DT)

```
1: p1 = rand(P), ||p2 - p1||_2 = min.            //Náhodný a nejbližší bod
2: Vytvoř hranu e = (p1p2)
3: p = arg min∀pi ∈σL(e) r 0 (k_i), k_i = (a, b, p_i), e = (a, b)
4: Pokud ∃p, prohod′ orientaci e ← (p2p1). Jdi na 3).
5: e2 = (p2, p), e3 = (p, p1)                    //Zbyvajici hrany trojuhelniku
6: AEL← e, AEL← e2, AEL← e3                      //Pridani 3 hran do AEL
7: DT ← e, DT ← e2, DT ← e3                      //Pridani 3 hran do DT
8: while AEL not empty:
9:    AEL → e, e = (p1p2)                        //Vezmi první hranu z AEL
10:   e = (p2p1)                                 //Prohod jeji orientaci
11:   p = arg min∀pi ∈L(e) r 0 (ki ), ki = (a, b, pi ), e = (a, b)
12:   if ∃p :                                    //Takovy bod existuje
13:         e2 = (p2, p), e3 = (p, p1)           //Zbyvajici hrany trojuhelniku
14:         DT ← e                               //Pridej hranu do DT ale ne do AEL
15:         add(e2,AEL,DT ), add(e3,AEL,DT )     //Pridej do DT i do AEL(?)
```

Algoritmus 3: Add (e = (a, b), AEL, DT)

```
1: Vytvoř hranu e 0 = (b, a)
2: if (e 0 ∈ AEL)
3: AEL → e 0            //Odstraň z AEL
4: else:
5: AEL ← e        //Přidej do AEL
6: DT ← (a, b).         //Přidej do DT
```

__Vrstevnice__

Vrstevnice zobrazují průsečné čáry vodorovných rovin v určité nadmořské výšce s terénem. Můžeme je rozdělit na vrstevnice základní, jež jsou vyobrazeny tenkou plnou čarou a vrstevnice hlavní, které zobrazujeme čárou silnější a jsou obvykle vykreslovány jako každá pátá nebo desátá vrstevnice.

Existují dvě základní metody tvorby vrstevnic – lineární konstrukce a nelineární konstrukce. Nelineární metoda je přesnější a reflektuje plynulou změnu terénu v realitě, ovšem je také výpočetně značně složitější a náročnější na implementaci

V našem případě byla zvolena metoda lineární konstrukce, která předpokládá, že spát terénu mezi dvěma body terénu, mezi kterými probíhá interpolace vrstevnic, je konstantní. Stejně tak je konstantní i rozestup vrstevnic.

Vrstevnice jsou generovány na každém trojúhelník, přičemž rozdíl z-souřadnice rozdělujeme na každé hraně lineárně. Pokud hrana nebo celý trojúhelník leží v rovině vrstevnic, vrstevnice neurčujeme.

Algoritmus:

1: Pro všechny hrany trojúhelníku určíme její vztah k rovině vrstevnice
2: if náleží rovině $(\mathrm{z}-z_i)\cdot(\mathrm{z}-\mathrm{z}_{\mathrm{i}+1}) = 0 \rightarrow e_i \,\epsilon\, \rho$
3: celá hrana je součástí vrstevnice, neřešíme
4: else if nenáleží rovině $(\mathrm{z}-z_i)\cdot(\mathrm{z}-\mathrm{z}_{\mathrm{i}+1}) > 0 \rightarrow e_i \notin \rho$
5: vrstevnice vůbec neprochází hranou, neřešíme
6: else if je průnikem roviny $(\mathrm{z}-z_i)\cdot(\mathrm{z}-\mathrm{z}_{\mathrm{i}+1}) < 0 \rightarrow e_i \cap \rho$
7: vypočteme průsečík vrstevnice s hranou

$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1$$

$$y = \frac{(y_2 - y_1)}{(z_2 - z_1)}(z - z_1) + y_1$$

Sklon

Sklon určujeme jako úhel mezi svislicí a normálou n trojúhelníku pro každý trojúhelník. Rovina trojúhelníku je určena vektory $\vec{u}$, $\vec{v}$.

$$n = (0,0,1)$$

$$n_t = \vec{u} \times \vec{v}$$

$$\varphi = arccos\frac{(n_t \cdot n)}{(|n_t| - |n|)}$$

Expozice

Expozici určujeme jako orientaci trojúhelníku ke světové straně.

$$A = arctan2\left(\frac{n_x}{n_y}\right)$$

Pro výpočet expozice byla využita metoda atan2, která automaticky určuje výsledný kvadrant, díky čemuž je výsledný úhel korektní.

Vstupní data

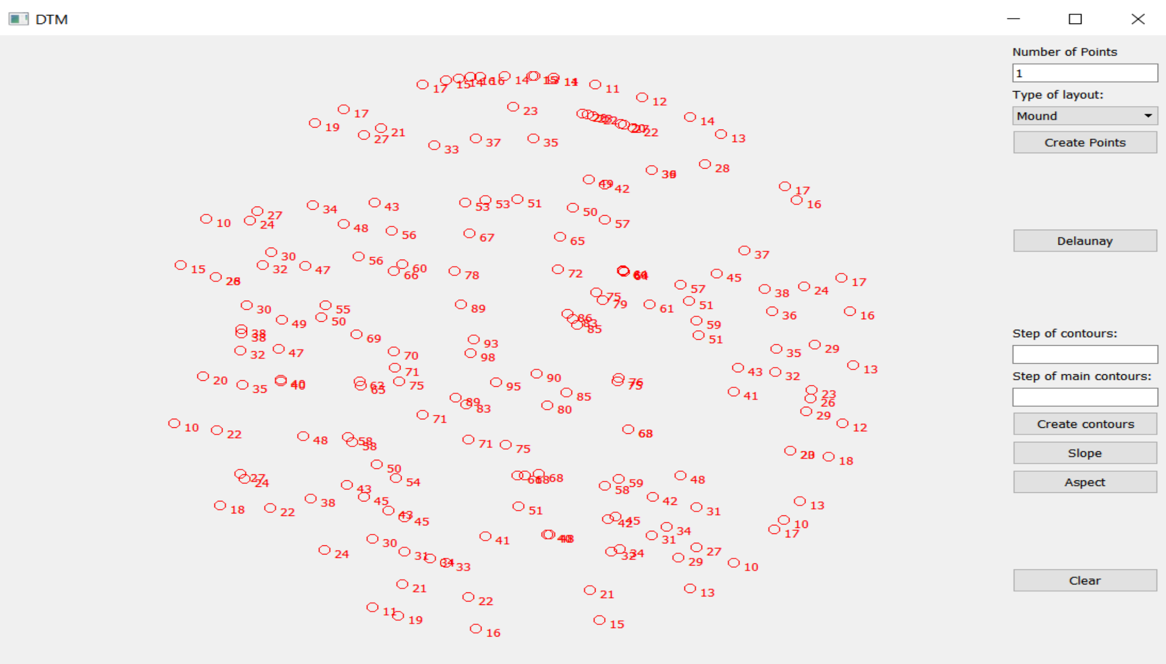
Množiny jsou generovány náhodně, lze volit s možností – náhodně, v mřížce (zde můžeme určit počet bodů, které chceme), případně zvolit možnost kupy (ang. Mound), kde je počet bodů dán v programu pevně.

Rovněž lze přidávat body manuálně kliknutím do aplikace.

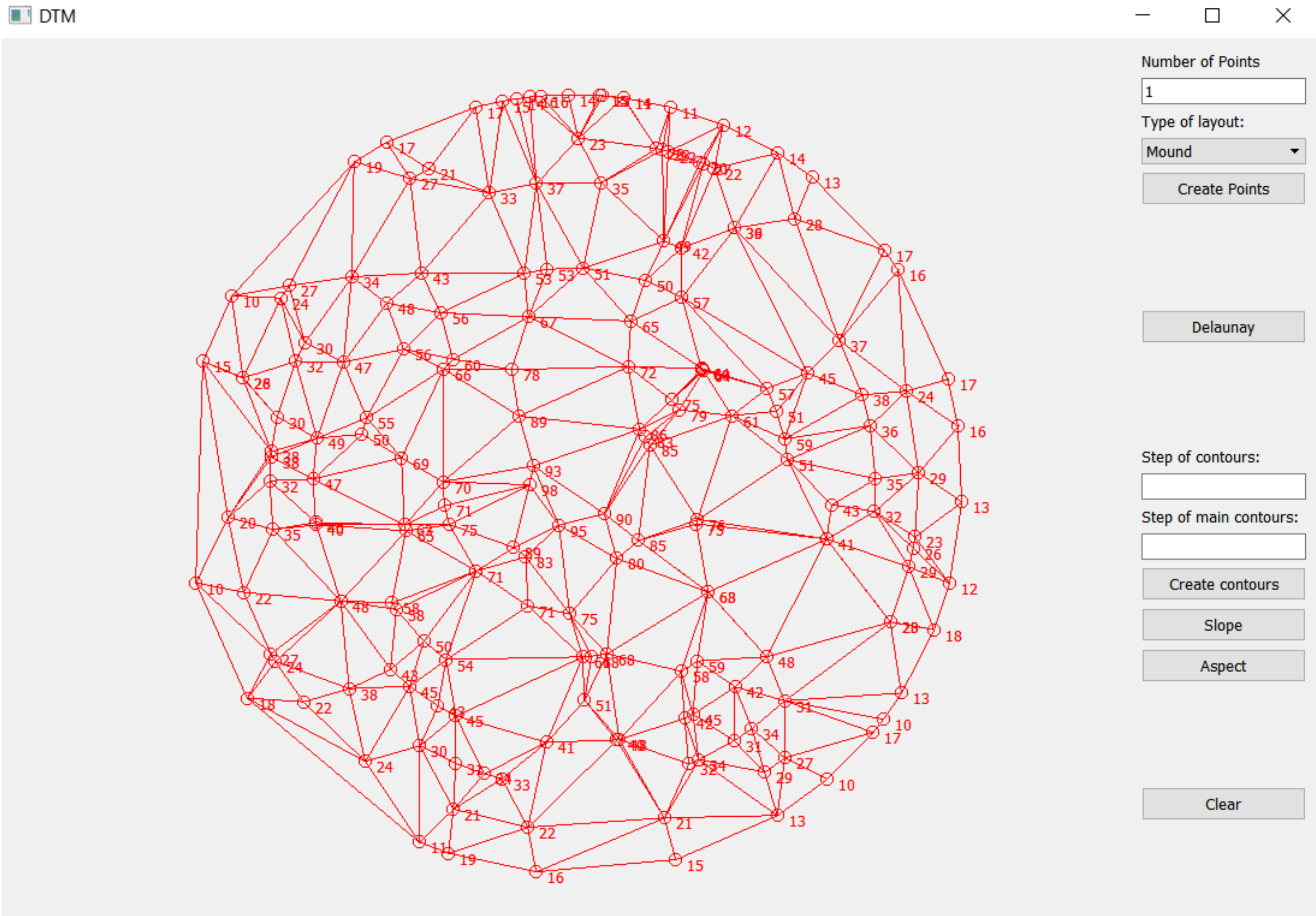
Výstupní data:

Aplikace zobrazuje vygenerované množiny s hodnotou z-souřadnice, dále vypočtenou Delaunay Tringulaci, vrstevnice s hlavními vrstevnicemi, jejichž krok si můžeme sami určit. Lze také určit sklon a expozici, jež jsou vyobrazeny ve stupních šedi.

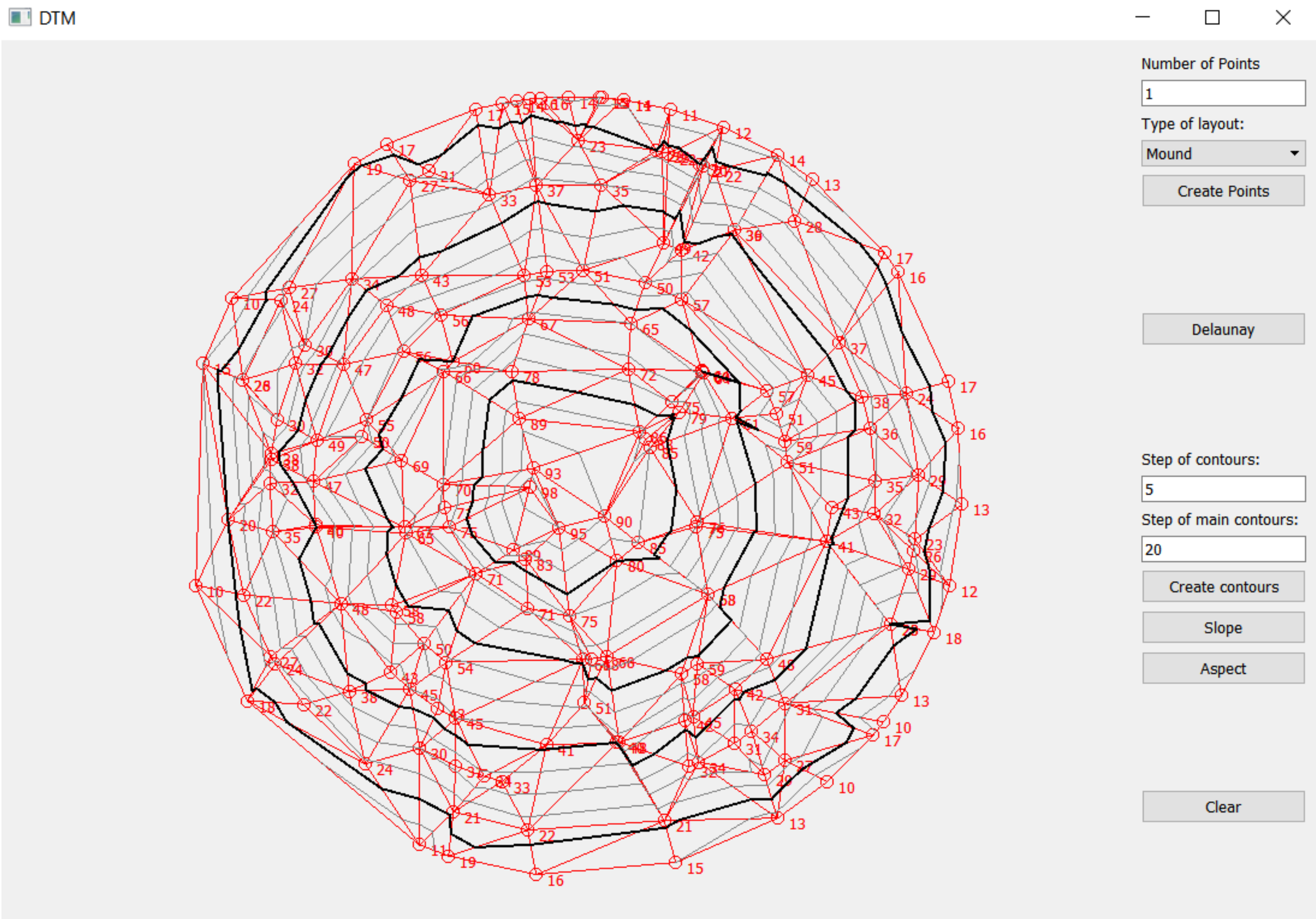
Printscreen:



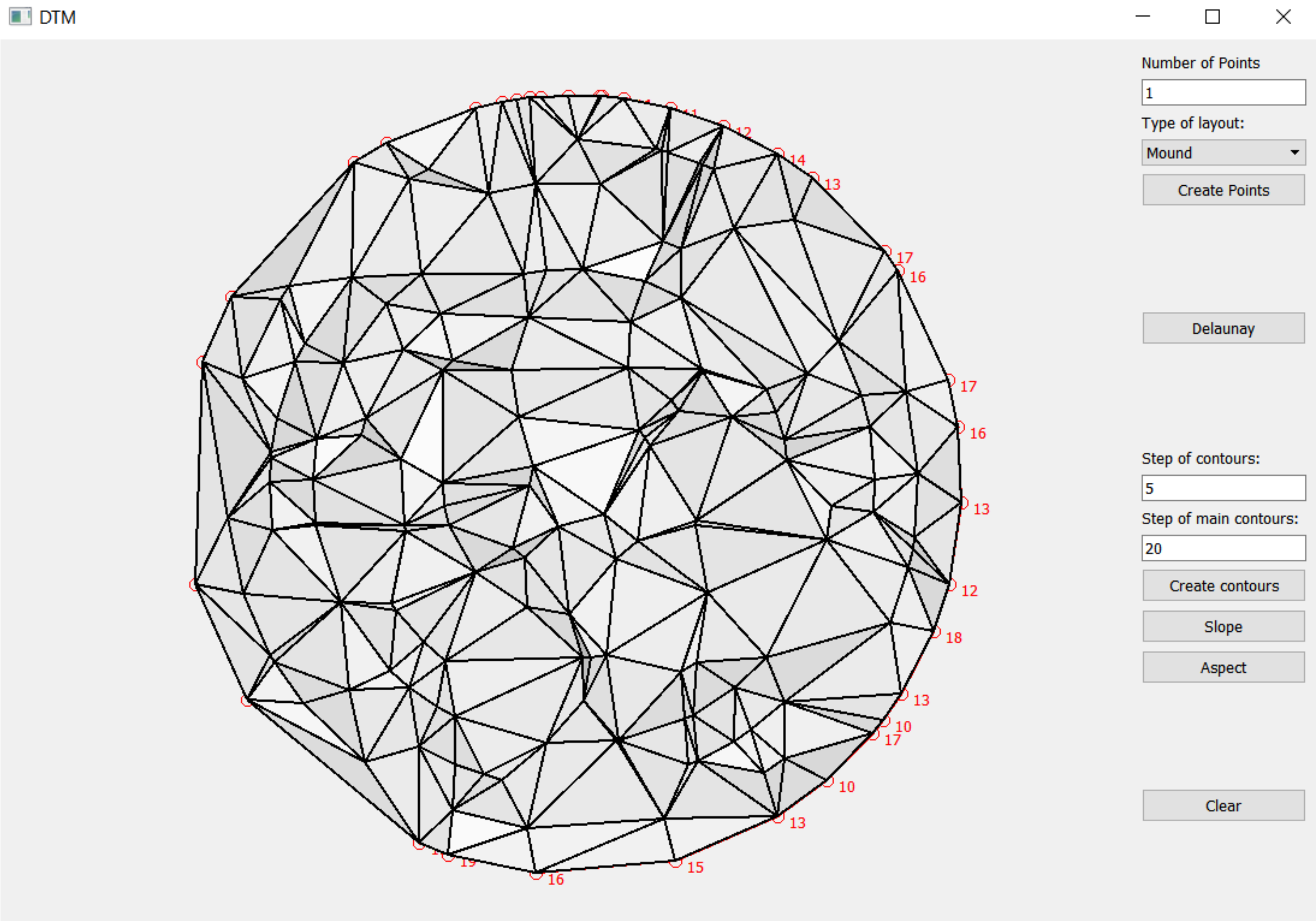
Obr.1 Vygenerovaná množina bodů.



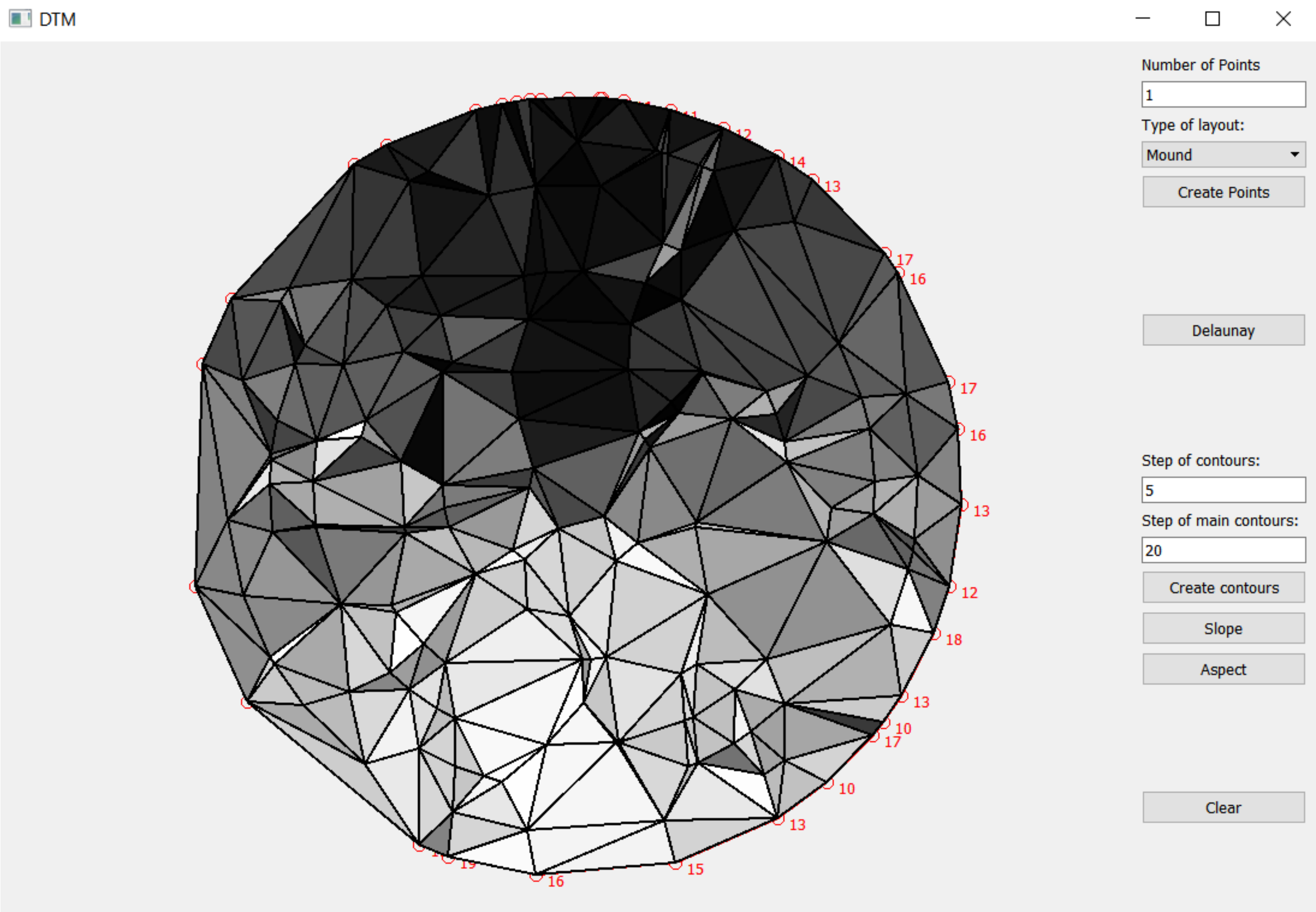
Obr.2 Vygenerovaná Delaunay Triangulace



Obr.3 Vygenerované vrstevnice



Obr.4 Vygenerovaná vizualizace sklonu.



Obr.5 Vygenerovaná vizualizace expozice

Zhodnocení:

Byl vytvořen program generující náhodné množiny bodů a následně pro ně počítající Delauney triangulaci, vrstevnice, expozici a sklon.

Aplikace dokáže vytvořit jak náhodné rozložení bodů, tak i rozložení simulující terénní kupu. Je možné zvolit vlastní rozestup vrstevnic, ať už základních tak hlavních. Sklon a expozice je zobrazen ve stupních šedi.

Algoritmus pracuje spolehlivě pro náhodné množiny bodů, při spuštění na kupě je tento útvar z vrstevnic a analýzy expozice jasně patrný. Problémy můžou nastat při výpočtu nad pravidelným gridem, jelikož DT je postavena na vytváření nejmenší opsané kružnice a volba není jednoznačná. Ovšem pro náhodné množiny se naše implementace osvědčila,