

České vysoké učení technické v Praze
Fakulta stavební



Algoritmy v digitální kartografii

Digitální model terénu a jeho analýzy

Bc. Petra Pasovská
Bc. David Zahradník

Obsah

1	Zadání	3
1.1	Údaje o bonusových úlohách	4
2	Popis a rozbor problému	4
3	Popis použitých algoritmů	5
3.1	Delaunay triangulace	6
3.1.1	Vlastnosti Delaunay triangulace	6
3.1.2	Implementace metody	6
3.2	Vrstevnice	7
3.2.1	Implementace metody	7
3.3	Sklon terénu	8
3.3.1	Implementace metody	8
3.4	Expozice terénu	8
3.4.1	Implementace metody	8
4	Informace o bonusových úlohách	9
4.1	Výběr barevných stupnic při vizualizaci sklonu a expozice	9
4.2	Algoritmus pro automatické generování terénních tvarů	9
4.2.1	Kupa	9
4.2.2	Údolí	9
4.2.3	Hřbet	10
4.2.4	Sedlo	10
4.2.5	Spočinek	10
5	Vstupní data	11
6	Výstupní data	11
7	Aplikace	12
8	Dokumentace	21
8.1	Třídy	21
8.1.1	Algorithms	21
8.1.2	Draw	22
8.1.3	SortByXAsc	23
8.1.4	Edge	23
8.1.5	QPoint3D	24
8.1.6	Triangle	24
8.1.7	Widget	24
9	Závěr a slovní zhodnocení	26
9.1	Delaunay triangulace	26
9.2	Vrstevnice	26
9.3	Sklon a expozice terénu	27
9.4	Výstupy aplikace	28

10	Náměty na vylepšení	29
10.1	Vrstevnice	29
10.2	Import dat	29
10.3	Sklon terénu	30
10.4	Tlačítka aplikace	30
11	Reference	31

1 Zadání

Níže uvedené zadání je kopie ze stránek předmětu.

Úloha č. 3: Digitální model terénu

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

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. navrhnete 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.

Zhodnot'te výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabunami 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 nekonverzní 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

1.1 Údaje o bonusových úlohách

V rámci bonusových úloh byl vytvořen výběr barevných stupnic pro vizualizaci sklonu a expozice. Dále jsou v rámci aplikace generované některé terénní tvary.

2 Popis a rozbor problému

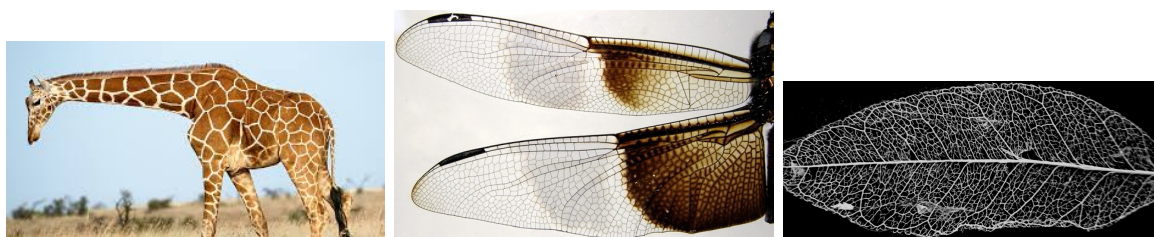
Hlavním cílem této úlohy je tvorba aplikace, která nad výškopisnými body vytvoří triangulaci, vrstevnice a vypočte sklon a expozici k světovým stranám.

Obecně triangulační algoritmy jsou nejvíce zkoumané algoritmy digitální kartografie v dnešní době. Slouží k různým účelům např. tvorba digitálního modelu terénu, plánování pohybu robotů, detekce otisků prstů. [Zdroj: 1]

Při triangulaci je kladen důraz na to, aby vytvořené trojúhelníky byly pokud možno rovnostranné. V případě, že takovéto trojúhelníky sestavíme, každý z vytvořených trojúhelníků musí co nejlépe reprezentovat hodnotu povrchu. Zároveň je nutné, aby byla produkována jednoznačná triangulace nezávisle na počátečním bodě či na orientaci množiny bodů. triangulace tyto podmínky obecně splňuje, přesto existují výjimky, kdy nemá Delaunay triangulace jednoznačné řešení. Tento stav může nastat pro určité množiny dat, např. pravoúhlý grid. [Zdroj: 2]

Delaunay triangulace je velmi podobná Dirichletově teselaci, která rozdělí body unikátní množinou polygonů, které jsou označovány jako Thiessenovy polygony či Voronoiovy diagramy. Voronoiovy diagramy jsou takové polygony, které vytvoří kolem všech bodů takové oblasti, že všechna místa uvnitř leží nejbližší k danému bodu.

Voronoiovy diagramy mají v dnešní době řadu využití. V meteorologii se využívají pro určení množství srážek v daném území, v astronomii pro studium galaxií, v molekulární biologii pro hledání tunelů v molekulách, v geografii pro sledování osídlení či pro plánování cest při pohybu robotů. Voroniov diagramy se vyskytují i v přírodě, můžeme je nalézt na kůži žirafy, na křídle vážky, na povrchu pouště Atacamy či na křídle vážky. [Zdroj: 3]



Obrázek 1: Voronoiovy diagramy v přírodě - na kůži žirafy, na křídle vážky a na žilkování listu

Tato dokumentace je však zaměřena na triangulaci. Existuje mnoho druhů triangulace v závislosti na geometrické konstrukci - Greedy triangulace, Minimum Weight Triangulation, Constrained triangulation, nejčastěji používaná je však Delaunay triangulace. Zjednodušeně si lze Delaunayho triangulaci představit tak, že zvolíme tři body, kterým

opíšeme kružnici. Pokud uvnitř kružnice neleží žádný další bod, vytvoří se trojúhelník. Pokud se uvnitř bod nachází, zvolí se jiné tři body. Tato triangulace je jednoznačná, pokud žádné čtyři body neleží na kružnici. Trojúhelníky vzniklé Delaunayho triangulací se nejvíce blíží rovnostranným trojúhelníkům.

V dnešní době se kromě analýzy prostorových dat využívá Delaunayho triangulace a Voronoiovy diagramy například v grafice. Takto vytvořené obrazy jsou nazývané Delaunayho rastry (The Delaunay Raster). Autoři takovýchto děl jsou schopni pomocí Delaunay triangulace sami vytvářet obrazy, případně převést existující obraz na triangulaci. [Zdroj: 4]



Obrázek 2: Použití Delaunayho triangulace v grafice - na vytvořeném novém obrázku lišky a na přepracování obrazu od Jana van Eycka

3 Popis použitých algoritmů

Existuje několik způsobů jak vytvořit triangulaci s různými kritérii. Obecně lze metody triangulace rozdělit na dvě skupiny. Jednou je skupina triangulací dle geometrické konstrukce. Mezi tyto triangulace se řadí Greedy triangulace, Delaunay triangulace, Minimum Weight triangulace, Constrained triangulace (triangulace s povinnými hranami) a datově závislé triangulace. Případně lze triangulace rozdělit dle použitých kritérií, tedy na lokálně optimální triangulace, globálně optimální triangulace a multikriteriálně optimalizované triangulace.

V této úloze jsme se zabývali Delaunay triangulací pomocí metody inkrementální konstrukce. Další metody přímé konstrukce Delaunay triangulace jsou např. metoda lokálníhoho

prohazování, inkrementálního vkládání, Divide and Conquer (rozděl a panuj) či Sweep Line (zametací přímka). Pro nepřímou konstrukci se využívají Voronoiho diagramy, v praxi však tento způsob není používán.

3.1 Delaunay triangulace

Delaunay triangulace je nejčastěji používanou triangulací při tvorbě digitálního modelu terénu. Delaunay triangulaci lze provádět v rovině i v prostoru.

Triangulace byla realizována pomocí metody inkrementální konstrukce. Tento algoritmus je založen na postupném hledání bodu, který k bodům hrany tvoří minimální opsanou kružnici. Každá hrana je orientovaná a bod se hledá pouze v její levé polorovině.

Je-li nalezen bod s výše uvedeným kritériem, vytvoří se dvě nové hrany, které jsou přidány do triangulace. Nenalezne-li se daný bod, prohodí se orientace hrany a hledání pokračuje.

Hrany, které nebyly zlegalizovány (nebyl k nim ještě nalezen třetí bod), jsou ukládány do struktury Active Edge List (AEL). Pokud k dané hraně byl nalezen třetí bod, hrana se ze struktury odstraní. Než je hrana vložena do struktury, kontroluje se, zda hrana už ve struktuře není s opačnou orientací. Pokud je, hrana se nevloží. Algoritmus probíhá do té doby, dokud není struktura AEL prázdná.

3.1.1 Vlastnosti Delaunay triangulace

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

3.1.2 Implementace metody

1. Nalezení pivota q s minimální souřadnicí X : $q = \min(x_i)$
2. Hledání nejbližšího bodu bodu: $\|p_1 - q\|_2 = \min$
3. Vytvoření první hrany: $e = (q, p_1)$
4. Hledání Delaunay bodu: $\underline{p} = \operatorname{argmin}_{p_i \in \sigma_L(e)} r'(k_i); k_i = (a, b, p_i); e = (a, b)$
 Při nenalezení Delaunay bodu změna orientace: $\nexists \underline{p} : e \leftarrow (p_1, q); \text{ go to 4.}$
5. Vytvoření zbývajících hran trojúhelníka: $e_2 = (p_1, \underline{p}); e_3 = (\underline{p}, q)$
6. Přidání hran trojúhelníka do DT: $DT \leftarrow e; DT \leftarrow e_2; DT \leftarrow e_3$
7. Přidání hran trojúhelníka do AEL: $AEL \leftarrow e; AEL \leftarrow e_2; AEL \leftarrow e_3$

8. Dokud není AEL prázdný proved':

Hledání Delaunay bodu k hraně z AEL (viz 4).

Pokud Delaunay bod existuje. $if \exists p$

Vytvoření zbývajících hran trojúhelníku: $e_2 = (p_1, \underline{p}); e_3 = (\underline{p}, q)$

Pokud nová hrana není v AEL, přidej.

3.2 Vrstevnice

Vrstevnice jsou půdorysným obrazem průsečných čar vodorovných rovin s terénem, přičemž jejich nadmořská výška je beze zbytku dělitelná zvoleným základem. Existují 4 druhy vrstevnic - základní, zvýrazněné, doplňkové a pomocné. Základní vrstevnice jsou vyobrazené tenkou křivkou a jejich rozestup je roven intervalu vrstevnic. Zvýrazněné vrstevnice jsou značeny tučně a obvykle jsou voleny jako pětinašobek základního intervalu. Doplňkové vrstevnice mají většinou poloviční či čtvrtinový interval a znázorňují se čárkovaně nejčastěji v místech plochého terénu. Pomocné vrstevnice slouží jen pro orientaci v místech, kde může docházet k podstatným změnám terénu (sesuvná území, lomy).

Při popisu vrstevnic existuje několik kartografických zásad. Pozice textu by měla symbolizovat terén, tedy hlava popisu směrem do kopce a pata do klesání. Dále by měl být popis zobrazen stejným způsobem jako popisovaná vrstevnice (barva, tučně apod). Z důvodu čitelnosti je nutné vrstevnici v místě popisu přerušit. Je vhodné, aby byl popis rovnoměrně rozmístěn po mapovém poli.

Existují dva základní způsoby konstrukce vrstevnic. U lineární interpolace je rozestup vrstevnic mezi dvěma body konstantní, tedy i spád. Při konstrukci vrstevnic hledáme průsečnici vodorovné roviny o výšce Z a rovinu trojúhelníka triangulace. V úloze byly vrstevnice konstruovány lineární interpolací.

3.2.1 Implementace metody

1. Pro všechny hrany trojúhelníku t : $\forall e_i \in t$:
2. Hrana náleží rovině vrstevnice z : $(z - z_i) \cdot (z - z_{i+1}) = 0 \rightarrow e_i \in \rho$
3. Hrana nenáleží rovině vrstevnice z : $(z - z_i) \cdot (z - z_{i+1}) < 0 \rightarrow e_i \notin \rho$
4. Hrana je průnikem roviny vrstevnice z : $(z - z_i) \cdot (z - z_{i+1}) < 0 \rightarrow e_i \cap \rho$

Výpočet polohových souřadnic:

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

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

Vytvoř hranu tvořící vrstevnici.

3.3 Sklon terénu

Skon terénu je definován jako úhel mezi normálovým vektorem (0,0,1) a normálovým vektorem roviny trojúhelníku. Pro výpočet je dostačující vypočítat směrové vektory v trojúhelníku. Do výpočtu vstupuje normálový vektor (0, 0, 1), jehož norma je rovna jedné a který v čitateli argumentu funkce arcus cosinus ponechá pouze Z část normálového vektoru.

3.3.1 Implementace metody

1. Pro všechny trojúhelníky triangulace: $\forall t_i \in DT$:

Výpočet normálového vektoru roviny trojúhelníku:

$$n_t = (u_y \cdot v_z - u_z \cdot v_y)^2 - (u_x \cdot v_z - u_z \cdot v_x)^2 + (u_x \cdot v_y - u_y \cdot v_x)^2$$

Kde:

$$u_x = \Delta x_2, x_1; u_y = \Delta y_2, y_1; u_z = \Delta z_2, z_1;$$

$$v_x = \Delta x_2, x_3; v_y = \Delta y_2, y_3; v_z = \Delta z_2, z_3;$$

2. Výpočet sklonu: $\varphi = \arccos \frac{n_z}{|n_t|}$

3.4 Expozice terénu

Expozice terénu je definována jako azimut k průmětu normálového vektoru roviny trojúhelníku do roviny x, y. Expozice má významný vliv na energetické poměry, neboť znázorňuje jednotlivé dopady slunečního záření, s čímž souvisí i výpar vody v oblasti.

3.4.1 Implementace metody

1. Pro všechny trojúhelníky triangulace: $\forall t_i \in DT$:

Výpočet x a y části normálového vektoru:

$$n_x = (u_y \cdot v_z - u_z \cdot v_y)$$

$$n_y = -(u_x \cdot v_z - u_z \cdot v_x)$$

Kde:

$$u_x = \Delta x_2, x_1; u_y = \Delta y_2, y_1; u_z = \Delta z_2, z_1;$$

$$v_x = \Delta x_2, x_3; v_y = \Delta y_2, y_3; v_z = \Delta z_2, z_3;$$

2. Výpočet expozice: $A = \text{atan2}(\frac{n_x}{n_y})$

Pro výpočet expozice byla použita funkce atan2, která automaticky určí i daný kvadrant, a tím je vypočtený úhel korektní směrnici.

4 Informace o bonusových úlohách

4.1 Výběr barevných stupnic při vizualizaci sklonu a expozice

V rámci této bonusové úlohy byly definovány barevné stupnice pro zobrazení sklonu a expozice. Pro sklon byly zvoleny odstíny šedi. Pro zobrazení šedé barvy v režimu RGB je typické to, že všechny tři složky mají stejné číslo. Bylo tedy snadné rozdělit interval 256 (jedná se o osmi bitovou hloubku) dle intervalu, ve kterém vychází sklon terénu.

Pro vyznačení expozice byly voleny syté barvy. Inspirací byla barevná škála v programu ArcGIS, přesto byly barvy zvoleny tak, aby byly asociativní se směrem, který reprezentují.

4.2 Algoritmus pro automatické generování terénních tvarů

V aplikaci je možnost automatického generování terénních tvarů. Na výběr je kupa, údolí, hřbet, grid, sedlo a spočinek. Přesto, že grid neoznačuje žádný terénní tvar, byl mezi možnostmi přidán z toho důvodu, abychom demonstrovali špatnou implementaci takového rozmístění bodů. Definice a anglická pojmenování jednotlivých útvarů byla použita ze stránky VÚGTK. [Zdroj: 5]

4.2.1 Kupa

Jedná se o takový terénní útvar, který je vyvýšený nad okolní terén. Běžně bývá tento tvar označován jako kopec či hora. V rámci aplikace je zvoleno pojmenování Hill.

Kupa byla generována klasickým způsobem jako elipsa, pro Z souřadnici bylo však navo-
lena, že hodnota postupně klesá. Pro vhodné zobrazení byl krok klesání zvolen 20.

$$X = X_0 + a * \cos(\phi)$$

$$Y = Y_0 + b * \sin(\phi)$$

$$Z = Z_0 - j * 20$$

4.2.2 Údolí

Jedná se o takový terénní útvar, který se vyznačuje protáhlým tvarem, jehož okraje jsou lemovány vyvýšeným terénem. V rámci aplikace je použito označení Valley.

Pro údolí byly vygenerovány přímky od nichž se v zadaném kroku vytváří nové přímky s konstantně stoupající hodnotou výšky. V uvedených vzorcích jsou p1 a p2 označení pro vzniklé přímky.

$$X_{p1} = X_0 + a$$

$$Y_{p1} = Y_0 + i * 10$$

$$Z_{p1} = Z_0 + a$$

$$X_{p2} = X_0 - a$$

$$Y_{p2} = Y_0 + i * 10$$

$$Z_{p2} = Z_0 + a$$

4.2.3 Hřbet

Hřbet je protáhlá vypuklá terénní plocha se zaoblenou vrcholovou částí. V aplikaci je pojmenován jako Ridge.

Hřbet byl generován podobným způsobem jako údolí, s tím rozdílem, že výška postupně nestoupá, ale klesá.

$$X_{p1} = X_0 + a$$

$$Y_{p1} = Y_0 + i * 10$$

$$Z_{p1} = Z_0 - a$$

$$X_{p2} = X_0 - a$$

$$Y_{p2} = Y_0 + i * 10$$

$$Z_{p2} = Z_0 - a$$

4.2.4 Sedlo

Sedlo je místo v krajině, které se nachází mezi dvěma vrcholy, z něhož na dvě strany svahu sestupují do údolí a na ostatní dvě stoupají směrem k vrcholům. U vysokých hor se sedlo často označuje jako průsmyk. V aplikaci je použito označení Saddle.

Generování sedla zahrnovalo vygenerování dvou vrcholů a dvou důlků. Podstatné bylo rozmístění. To se regulovalo zadanými pevnými souřadnicemi vrcholů, případně děr. následně byly kolem těchto středu generovány elipsy s klesající/rostoucí výškou.

Tvorba vrcholů:

$$X = X_0 + i * rand\%(70) * cos(j * \phi) + rand\%(20)$$

$$Y = Y_0 + i * rand\%(70) * sin(j * \phi) + rand\%(20)$$

$$Z = Z_0 - i * 50$$

Tvorba děr:

$$X = X_0 + i * rand\%(70) * cos(j * \phi) + rand\%(20)$$

$$Y = Y_0 + i * rand\%(70) * sin(j * \phi) + rand\%(20)$$

$$Z = Z_0 + i * 50$$

4.2.5 Spočinek

Spočinek je vodorovná nebo mírně skloněná část terénního reliéfu na svahovém hřbetu nebo svahu. V aplikaci lze spočinek vygenerovat názvem Col Topographic.

Tvorba spočinku se náhodně vytváří proto, že v kódu je tvorba nových bodů pouze v jednom kvadrantu a vzhledem k pravděpodobnosti blízkosti některých bodů se při generování vytváří v některých situacích spočinek. Tato tvorba je však zcela náhodná a

neřízená, autoři si za to nenárokují žádné body z této úlohy, spočinek byl v úloze ponechán z toho důvodu, že na něm lze demonstrovat nefunkčnost tvorby vrstevnic, ale funkčnost vytvořeného sklonu a expozice.

Tvorba spočinku:

$$X = X_0 + i * rand\%(100) * cos(j * \phi) + rand\%(50)$$

$$Y = Y_0 + i * rand\%(150) * sin(j * \phi) + rand\%(50)$$

$$Z = Z_0 - i * 50$$

5 Vstupní data

V aplikaci lze generovat jednotlivé útvary terénu a nebo lze importovat data ve formátu .txt. Automatická tvorba bodů kurzorem byla odstraněna.

Vzor vstupních dat ve formátu .txt

748720.120 1048636.300 306.100

748720.750 1048636.130 305.940

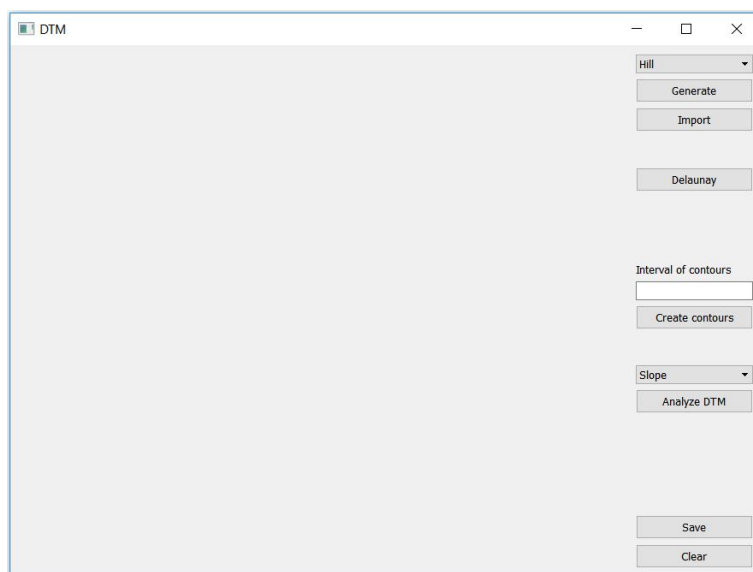
748721.600 1048635.700 306.080

748721.610 1048628.180 305.570

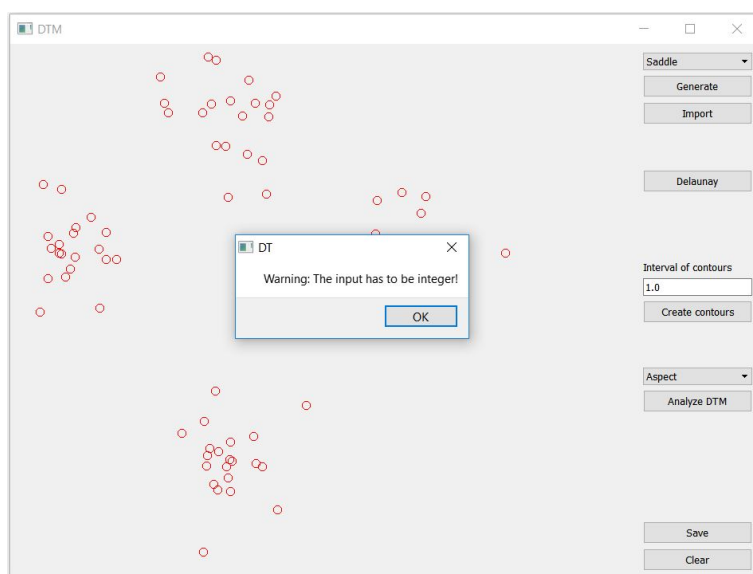
6 Výstupní data

Výstupem je grafická aplikace, která na vstupních datech vytvoří Delaunay triangulaci, se zvoleným krokem vykreslí vrstevnice a následně je možné zobrazit sklon či expozici terénu.

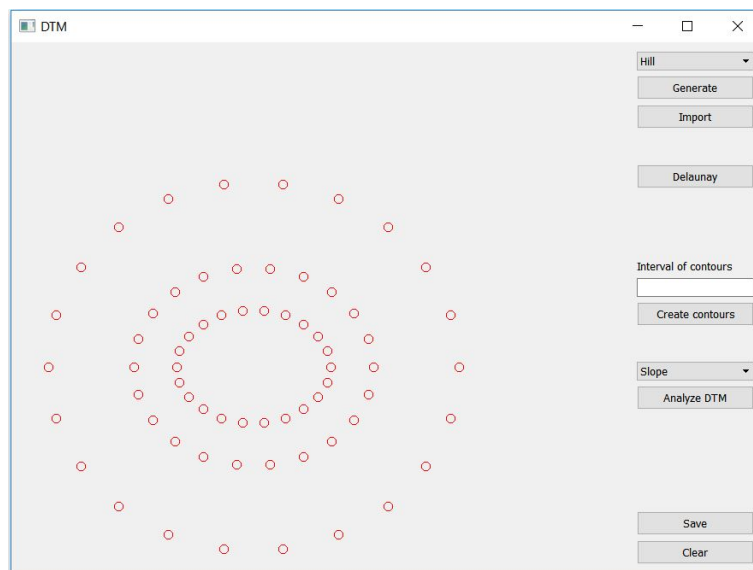
7 Aplikace



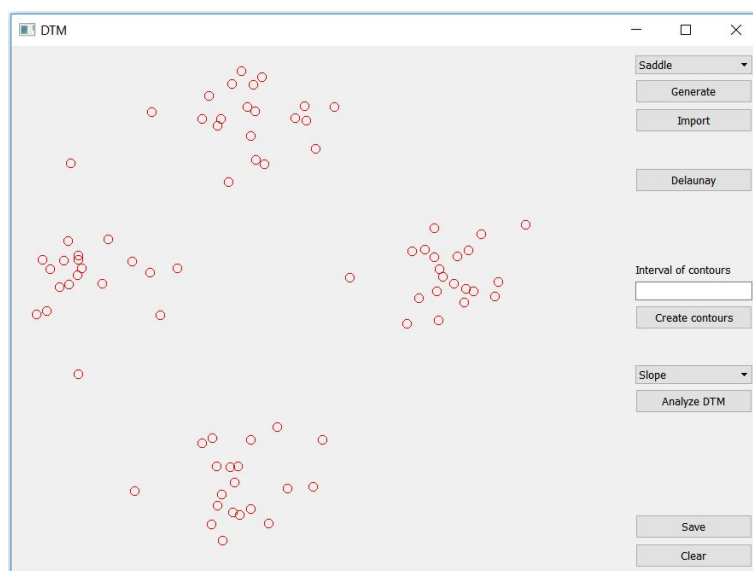
Obrázek 3: Vstupní vzhled aplikace



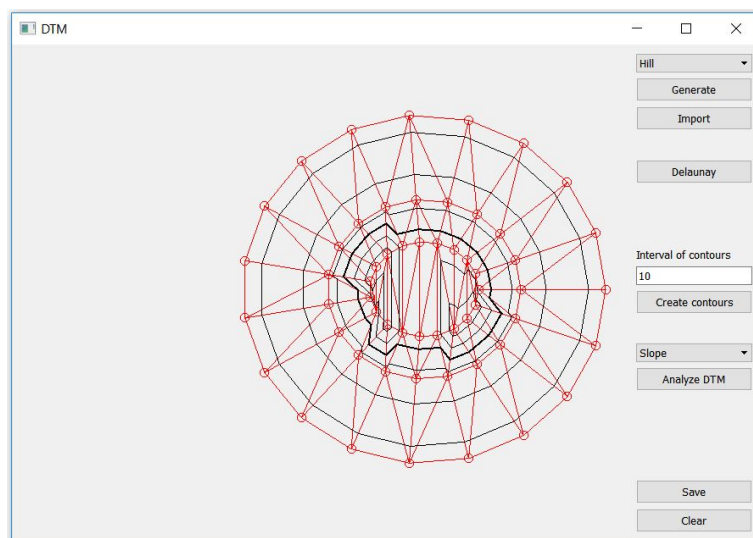
Obrázek 4: Varování při nesprávném vstupu intervalu souřadnic (zde 1.0, je nutné zadávat ve formátu celého čísla)



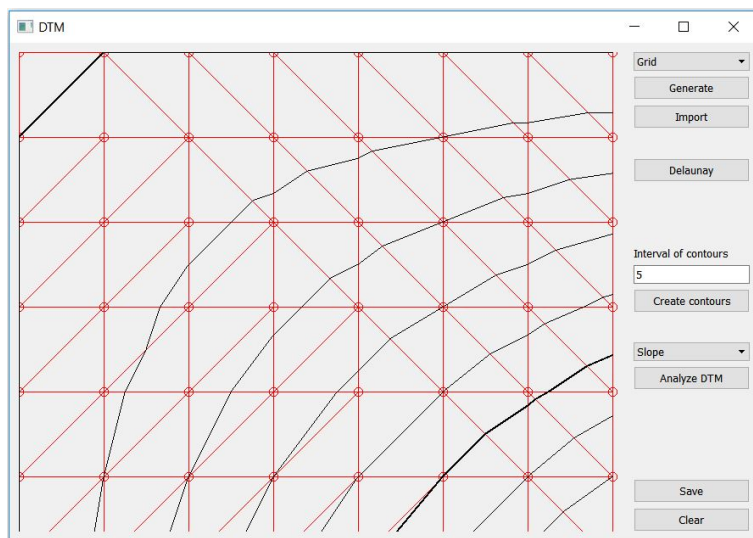
Obrázek 5: Vygenerované body znázorňující kupu



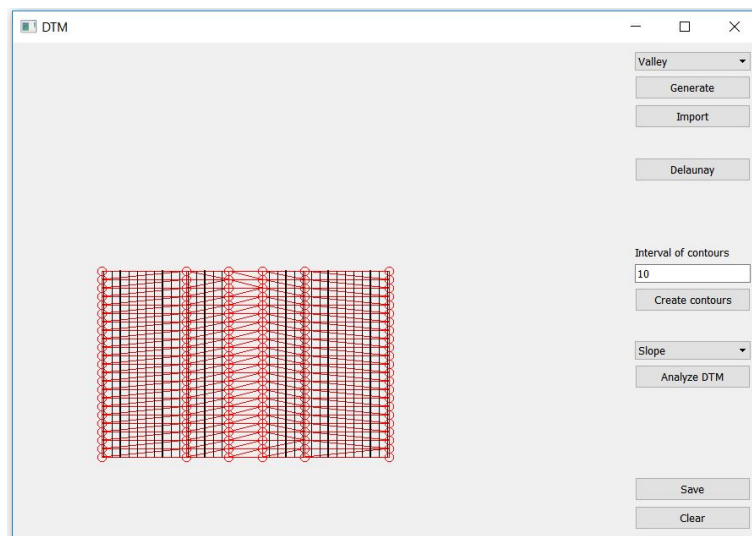
Obrázek 6: Vygenerované body znázorňující sedlo



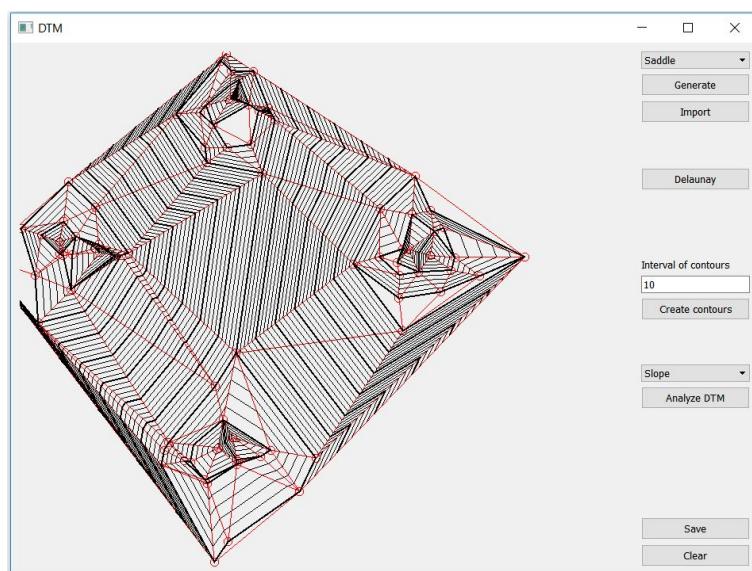
Obrázek 7: Špatně vykreslené vrstevnice kupy



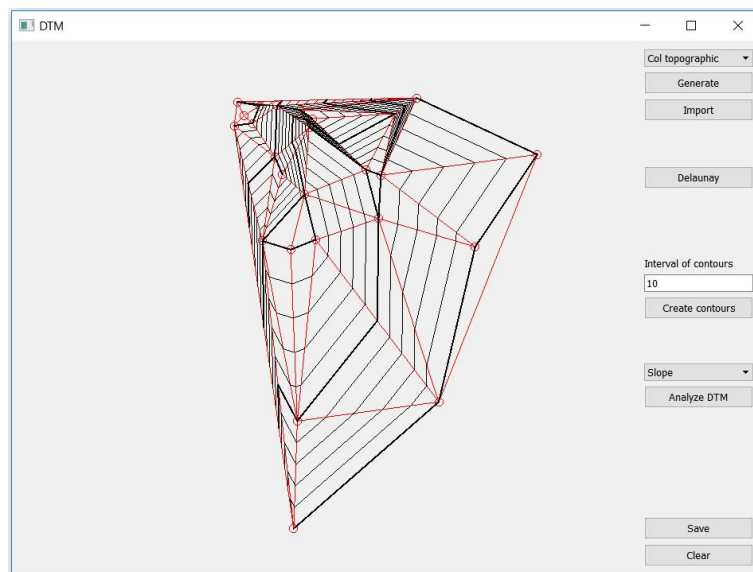
Obrázek 8: Vrstevnice pro grid jsou vykresleny korektně



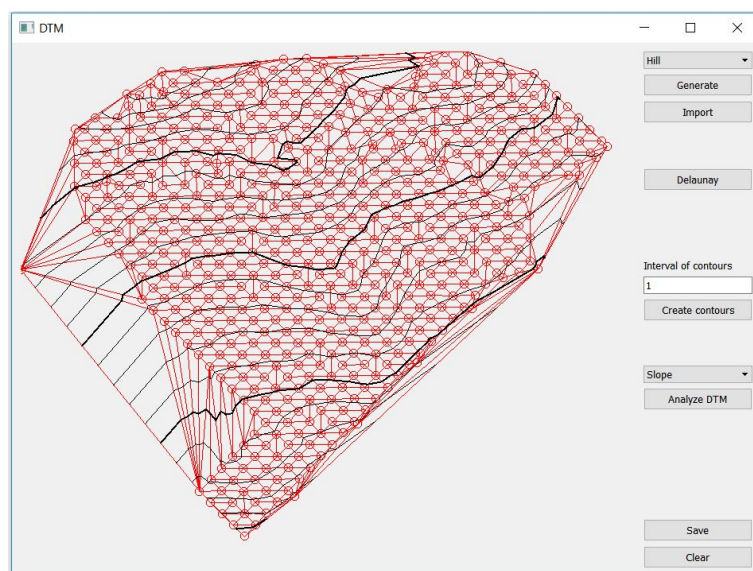
Obrázek 9: Vrstevnice pro údolí vykresleny korektně



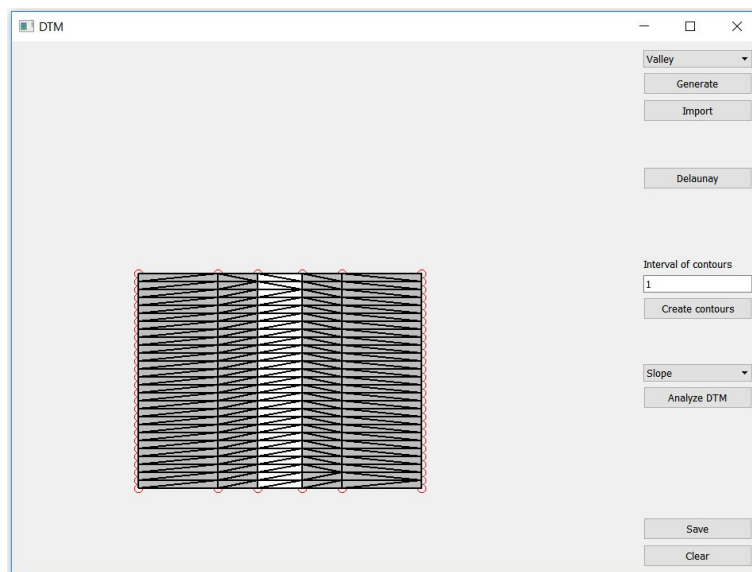
Obrázek 10: Nekorektně vykreslené vrstevnice pro sedlo



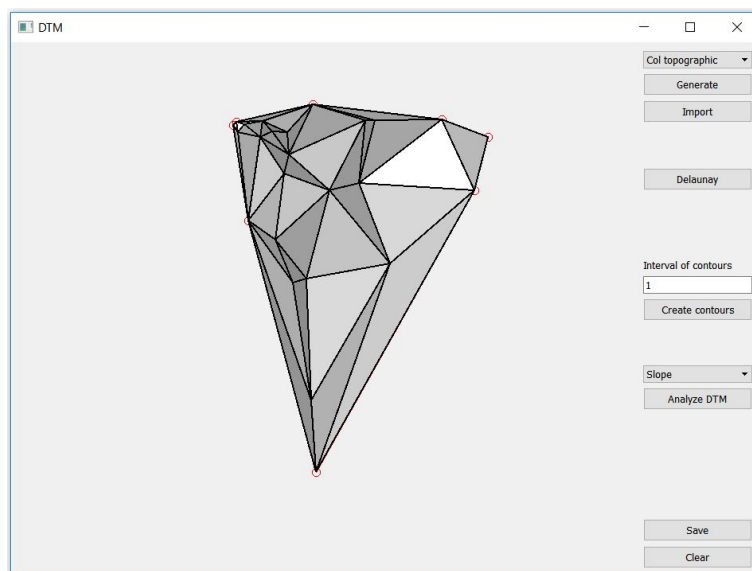
Obrázek 11: Nekorektně vykreslené vrstevnice pro spočinek



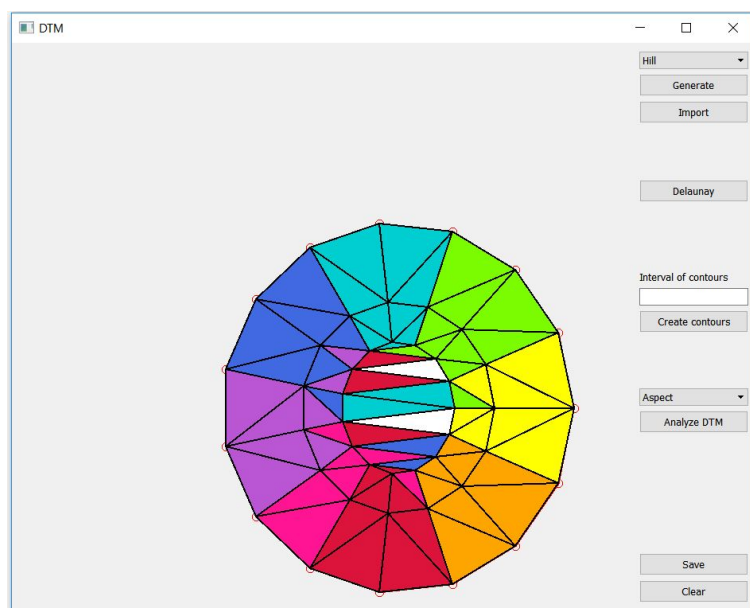
Obrázek 12: Vygenerované vrstevnice pro importovaná data



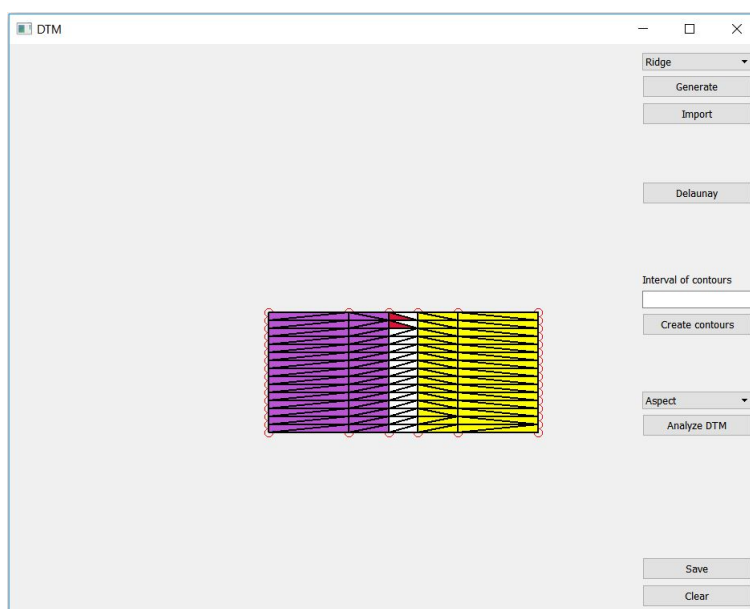
Obrázek 13: Zobrazení sklonu pro údolí korektní



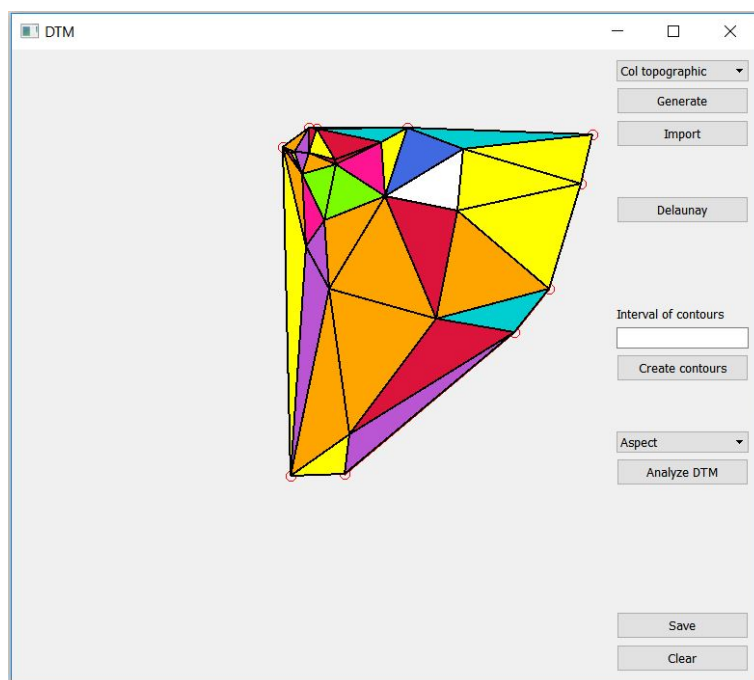
Obrázek 14: Oblast spočinku vykreslena správně



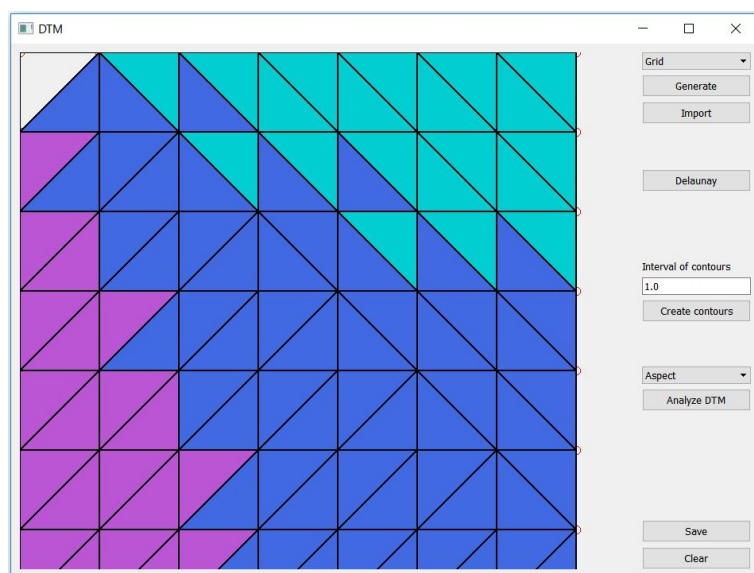
Obrázek 15: Expozice pro kupu vykreslena téměř správně až na oblasti, kde selhávaly i vrstevnice



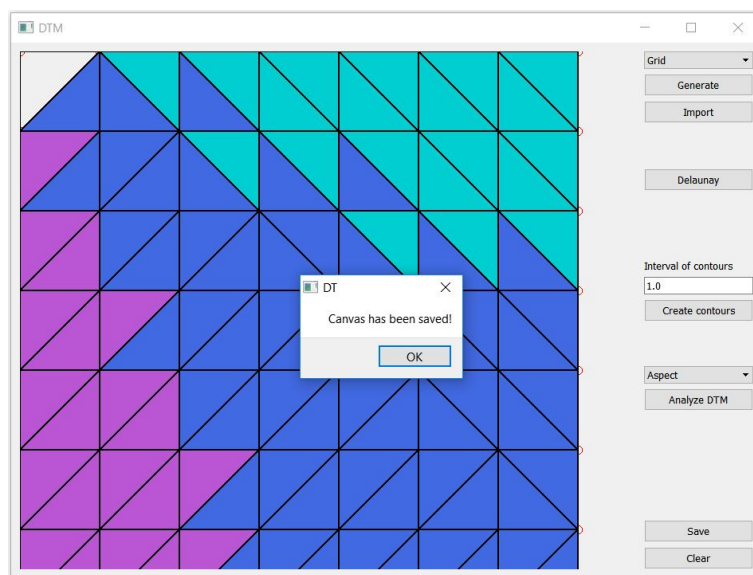
Obrázek 16: Expozice pro hřbet nezobrazeny korektně



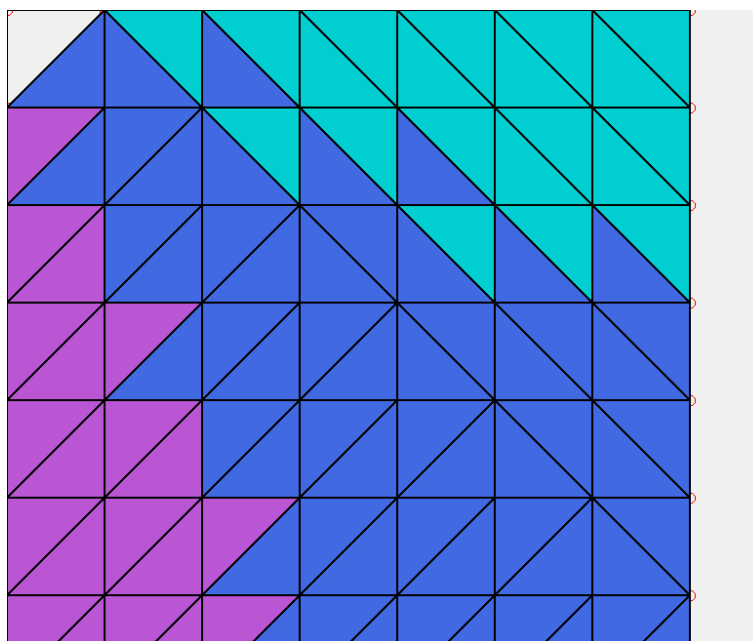
Obrázek 17: Korektně vytvořená expozice pro spočinek



Obrázek 18: Expozice pro grid nezobrazeny korektně



Obrázek 19: Ukládání zájmové oblasti



Obrázek 20: Pro ukázkou vyexportovaná zájmová oblast

8 Dokumentace

8.1 Třídy

8.1.1 Algorithms

Třída Algorithms obsahuje několik metod. Metody jsou určeny pro výpočty použitých algoritmů.

double distance(QPoint3D p1, QPoint3D p2)

Metoda, jejíž návratová hodnota je typu double, vrací velikost spojnice mezi dvěma body.

TPosition getPointLinePosition(QPoint &q, QPoint &a, QPoint &b)

Tato metoda slouží k určení pozice bodu q vůči linii tvořené body a, b. Výstupem metody je LEFT, RIGHT nebo ON.

double getCircleRadius(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3, QPoint3D &c)

Metoda jejíž návratová hodnota je typu double, vrací velikost poloměru kružnice tvořené třemi vstupními body.

int getNearestPoint(QPoint3D &p, std::vector<QPoint3D>&points)

Tato metoda slouží k nalezení indexu nejbližšího bodu k bodu p.

int getDelaunayPoint(QPoint3D &s, QPoint3D &e, std::vector<QPoint3D>&points)

Tato metoda slouží k nalezení indexu bodu, který splňuje Delaunay vlastnosti.

std::vector<Edge> DT(std::vector<QPoint3D>&points)

Metoda vytváří nad vektorem bodů Delaunay triangulaci, která je reprezentována jako vektor hran.

QPoint3D getContourPoint(QPoint3D &p1, QPoint3D &p2, double z)

Metoda vypočte průsečík hrany, tvořenou 3D body p1 a p2, a rovinou definovanou Z souřadnicí.

std::vector<Edge> createContours(std::vector<Edge>&dt, double z_{min}, double z_{max}, double z_{step})

Metoda z triangulace dt v zadaném intervalu v ose Z $< z_{min}; z_{max} >$ s intervalem vrstevnic dz vrátí vektor hran definujících vrstevnice.

double getSlope(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3)

Tato metoda slouží k výpočtu hodnoty sklonu trojúhelníku definovanému 3D body p1, p2 a p3.

double getAspect(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3)

Tato metoda slouží k výpočtu hodnoty expozice trojúhelníku definovanému 3D body p1, p2 a p3.

std::vector<Triangle>analyzeDTM(std::vector<Edge >&dt)

std::vector<QPoint3D>generateHill()

Tvorba kupy.

std::vector<QPoint3D>generateValley()

Tvorba údolí.

std::vector<QPoint3D>generateMountains()

Tvorba hřbetu.

std::vector<QPoint3D>generateGrid()

Tvorba gridu.

std::vector<QPoint3D>generateSaddle()

Tvorba sedla.

std::vector<QPoint3D>generateCol()

Tvorba spočinku.

8.1.2 Draw

Třída Draw obsahuje několik metod. Metody jsou určeny pro generování a vykreslování proměnných.

void paintEvent(QPaintEvent *e)

Metoda slouží k vykreslení vytvořených, generovaných bodů a zobrazení výsledků použitých algoritmů.

void mousePressEvent(QMouseEvent *e)

Metoda uloží bod se souřadnicemi místa kliknutí v zobrazovacím okně.

void clearDT()

Metoda slouží k vymazání proměnných a k překreslení

void clearPoints()

Metoda slouží k vymazání bodů.

void setPoints(std::vector<QPoint3D >points_)

Metoda slouží pro převod bodů do vykreslovacího okna.

std::vector<QPoint3D >& getPoints()

Metoda slouží pro převod bodů z vykreslovacího okna.

std::vector<Edge >& getDT()

Metoda slouží pro převod Delaunay triangulace z vykreslovacího okna.

void setDT(std::vector<Edge>&dt_)

Metoda slouží pro převod Delaunay triangulace do vykreslovacího okna.

void setContours(std::vector<Edge>&contours_)

Metoda slouží pro převod vrtevníc do vykreslovacího okna.

void setMainContours(std::vector<Edge>&mainContours_)

Metoda slouží pro převod vrtevníc do vykreslovacího okna.

void setDTM(std::vector<Triangle>&dtm_)

Metoda slouží pro převod trojúhelníku triangulace a jeho informací o sklonu a expozici terénu

void importPolygons(std::string &path, std::vector<QPoint3D >&points, QSizeF &canvas_size, double &min_z, double &max_z)

Metoda slouží pro import vrstevnic, z cesty path naplní vektor points body a uloží hodnotu s minimální a maximální souřadnicí. Proměnná canvas_size slouží k vykreslení v rozsahu importovaných dat.

void setSlope(bool slope_)

Metoda slouží jako podmínka TRUE/FALSE pro vykreslení sklonu terénu.

void setAspect(bool aspect_)

Metoda slouží jako podmínka TRUE/FALSE pro vykreslení expozice terénu.

8.1.3 SortByXAsc

Třída SortByXAsc slouží k porovnání souřadnic v ose x.

bool operator()(QPoint &p1, QPoint &p2)

Přetížený operátor () vrátí bod s větší souřadnicí x z dvojice bodů.

8.1.4 Edge

Edge(QPoint3D &start, QPoint3D &end)

Třída Edge je konstruována ze dvou 3D bodů, počátek a konec hrany. Třída slouží k uložení hrany triangulace a nebo vrstevnic.

QPoint3D &getS()

Metoda vrátí počáteční bod hrany.

QPoint3D & getE()

Metoda vrátí koncový bod hrany.

void switchOr()

Metoda prohodí orientaci hrany.

8.1.5 QPoint3D

QPoint3D(double x, double y, double z_)

Třída QPoint3D je odvozena z třídy QPointF a složí k uložení bodu s informací o výšce.

double getZ()

Metoda vrátí výšku bodu.

void setZ(double z_)

Metoda nastaví výšku bodu.

8.1.6 Triangle

Triangle(QPoint3D &p1_, QPoint3D &p2_, QPoint3D &p3_, double slope_, double aspect_)

Třída QPoint3D složí k uložení trojúhelníku definovaného body p1, p2, p3 a jeho informací o sklonu a expozici.

QPoint3D getP1()

Metoda vrátí první bod trojúhelníku.

QPoint3D getP2()

Metoda vrátí druhý bod trojúhelníku.

QPoint3D getP3()

Metoda vrátí třetí bod trojúhelníku.

double getSlope()

Metoda vrátí sklon trojúhelníku.

double getAspect()

Metoda vrátí expozici trojúhelníku.

8.1.7 Widget

void on_pushButton_clicked()

Při stisknutí tlačítka Denaulay se zavolá metoda třídy Algorithms DT a výsledek se zobrazí v okně.

void on_pushButton_3_clicked()

Při stisknutí tlačítka Clear se zavolá metoda třídy Draw clearDT.

void on_pushButton_2_clicked()

Při stisknutí tlačítka Create Contours se zavolá metoda třídy Algorithms createContours a výsledek se zobrazí v okně.

void readNumber

Ošetření při neplatném vstupu intervalu vrstevnic.

void on_pushButton_4_clicked()

Při stisknutí tlačítka AnalyzeDTM se zavolá metoda třídy Algorithms analyzeDTM a výsledek se zobrazí v okně.

void on_pushButton_5_clicked()

Při stisknutí tlačítka Generate a výběru z comboboxu se vygenerují body terénu.

void on_pushButton_6_clicked()

Při stisknutí tlačítka Import, se otevře okno pro výběr importovaných dat.

void on_Save_clicked()

Při stisknutí tlačítka Save, se otevře okno pro volbu místa uložení vytvořeného obrázku a následně je obrázek uložen.

9 Závěr a slovní zhodnocení

Byla vytvořena aplikace, která nad importovanými body vytvoří Delaunay triangulaci, vykreslí vrstevnice, sklon a expozici terénu. Součástí závěru je slovní zhodnocení činnosti algoritmu. Grafické ukázky aplikace jsou v kapitole 7 - Aplikace.

9.1 Delaunay triangulace

Vzhledem k implementaci triangulace pro výběr trojúhelníku s vyhledáním nejmenší opsané kružnice, algoritmus selhává pro body na mřížce. Triangulace je pro tyto body nejednoznačná a další výpočty nefungují. Obecně se ví, že Delaunayho triangulace nemá pro body na mřížce jednoznačné řešení. Tento problém by se mohl vyřešit vložením povinných hran. Zároveň aplikace selhává pro některé terénní tvary, tento problém by se dal taktéž vyřešit vložením povinných hran.

V aplikaci bohužel nelze nadefinovat povinné hrany, které jsou důležité pro terénní tvary, například pro terénní hranu či propast. Proto algoritmus není použitelný pro data s těmito typy terénních tvarů. Tvorba povinných hran by přitom výrazně zkvalitnila aplikaci, neboť by uživatel získával věrohodnější výstupy. S tímto problémem souvisí i to, že jsou následně chybně vykresleny vrstevnice. Zároveň v rámci aplikace není možné vymazávat neexistující hrany při vytvořeném nekonvexním polygonu oblasti. Díky tomu je triangulace tvořena i na místech, kde nebyly podrobné body a chybně jsou v těchto místech vytvářeny vrstevnice, sklon a expozice.

V aplikaci též nelze nastavit zájmovou oblast, proto je nutné před výpočtem algoritmu odstranit body, nad kterými nechceme provádět triangulaci. Při importu není řešen souřadnicový systém nahanraných dat. Pro nahrávání souřadnic se zápornou hodnotou nejsou body řádně vykresleny. Tento problém by se dal vyřešit transformací vstupních bodů volbou souřadnicového systému, ve kterém jsou data uložena v textovém souboru a takového souřadnicového systému, který data vhodně zobrazí ve vykreslovacím okně. V této úloze se však primárně zabýváme tvorbou digitálního modelu, není tedy na transformaci brán ohled a mezi testovanými soubory jsou jen takové body, které se v rozhraní správně vykreslí.

Aplikace je tedy užitečná pro výškopisná data v terénu bez povinných hran a kde data nepotřebují být upravována.

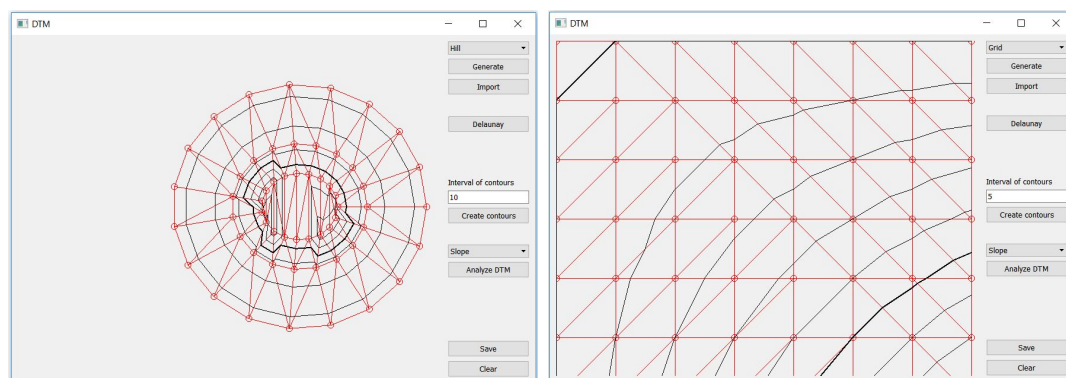
9.2 Vrstevnice

Pro vykreslení vrstevnic byla použita lineární interpolace, která má svá úskalí. Jelikož pro výpočet vrstevnic není řešené vyhlazení vrstevnic, výsledek není vhodný ke chlubení. Pro kvalitní výsledek by bylo vhodnější použít morfologickou interpolaci, která předpokládá plynulou změnu spádu terénu mezi jednotlivými body. Není pro ni však definován žádný exaktní postup, tudíž nelze naimplementovat. Pro morfologickou interpolaci je typické to, že je tvořena "od oka". Je typická pro mapy menšího měřítka, kde je redukován počet

zaměřených bodů a využívá zákonitosti terénních tvarů.

Pro interval vrstevnic je nutné, aby byl celočíselný. Při vkládání intervalu, který je například typu double nebo zůstane prázdné pole, je uživatel upozorněn na chybu a vykreslení vrstevnic je ukončeno. Tento krok bylo nutné učinit z toho důvodu, že aplikace v opačném případě padala a bylo zdlouhavé při zapomenutí nastavení volby kroku stále aplikaci spouštět znovu. Tato část kódu byla nakonec ošetřena jako výjimka, která zabrání pádu aplikace.

V rámci aplikace byly zvýrazněny hlavní vrstevnice pro lepší přehlednost.



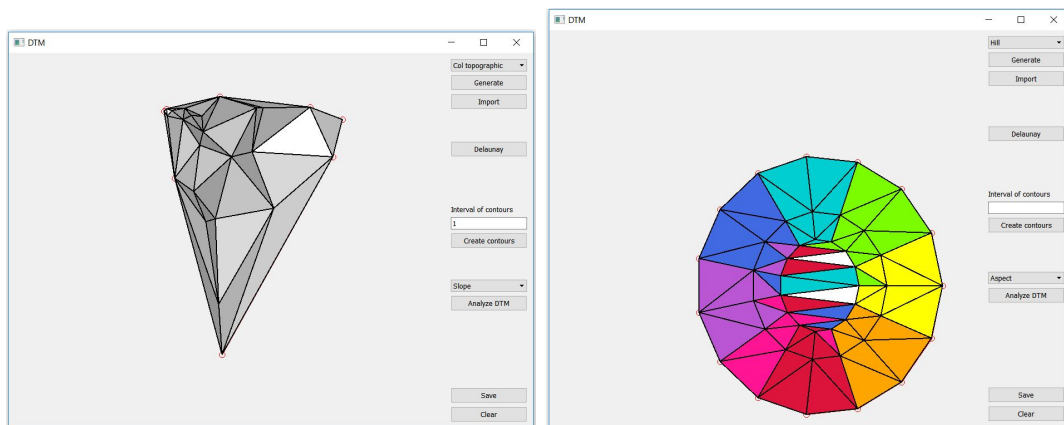
Obrázek 21: Vrstevnice pro kupu vykresleny nekorektně, pro grid korektně

9.3 Sklon a expozice terénu

Při výpočtu sklonu a expozice terénu vzniká problém v rovinatém území. Při výpočtu vznikají zaokrouhlovací chyby, které se promítnou do sklonu a expozice. Proto se terén jeví uživateli nerovinný.

Při vykreslení sklonu terénu byly pro ilustraci zvoleny odstíny šedi. Sklon nabývá hodnot od 0 do 180 °. Vykreslení expozice bylo rozděleno do osmi směrů - S, SV, V, JV, J, JZ, Z, SZ. Barvy znázorňující jednolivé směry byly voleny asociativně s určitou symbolikou. Pro sever byla zvolena chladná modrá, pro jih naopak teplá červená barva. Pro východ byla zvolena žlutá barva reprezentující slunce a pro západ byla fialová, aby byla pokryta celá škála barev. Pro směry tvořené kombinací jiných směrů byla vybrána taková barva, pro kterou lze vidět určitý přechod obou barev. Pro kvalitnější znázornění expozice a jednotlivých přechodů by jistě bylo vhodnější použít více intervalů, neboť v prostředí Qt Creatoru můžeme vytvořit obrovské množství různých odstínů. Přesto by byl výsledek poté nepřehledný a uživatel by se v tom nemusel zcela orientovat.

Bylo by vhodné do aplikace umístit legendu expozice, neboť barvy samy o sobě uživateli nic neřeknou.



Obrázek 22: Sklon pro spočinek vykreslen korektně, expozice pro kupu téměř správně, až na oblasti, kde selhávaly i vrstevnice

9.4 Výstupy aplikace

Uživatel má v aplikaci možnost uložit si vytvořený model nainportovaných souřadnic, Delaunay triangulací, vrstevnic, sklonu či expozice. Uživatel je vždy tázán na místo, kam nově vzniklý obrázek uložit. V případě úspěšného uložení je uživateli zobrazeno vyskakovací okno s informací o úspěšném uložení.

V rámci bonusových úloh byly také vytvořeny některé terénné útvary, například kupa, sedlo nebo údolí. Při tvorbě hřbetu došlo ke špatné implementaci a shodou náhod byl napsán takový kód, který vygeneruje spočinek. Generování spočinku je ovlivněno celou řadou proměnných, které jej generují. Záleží na zvoleném množství bodů, zadané generované vzdálenosti atd. Tato tvorba je také ovlivněna tím, že body se generují pouze v jednom kvadrantu, čímž se zvýší šance na to, aby byly poblíž sebe body se stejnou výškou a vytvořily tím spočinek. I přesto, že implementace spočinku je zcela náhodná, tudíž není korektní a občas se může stát, že spočinek nebude vygenerován, byla tato volba v kódu ponechána z toho důvodu, aby bylo možné na spočinku demonstrovat tvorbu vrstevnic a výpočet sklonu.

Co se týče výstupů aplikace, pro některé útvary funguje kód korektně, pro jiné se vyskytují chyby. Ukázky jednotlivých výsledků lze nalézt v kapitole 7 - Aplikace. Níže následuje shrnutí jednotlivých metod a útvarů.

Při detailnějším zkoumání vrstevnic, pro údolí a hřbet jsou vrstevnice znázorněny korektně. Pro znázornění vrstevnic kupy však dochází k chybám a vrstevnice nejsou korektně zobrazeny. Důvodem je to, že body jsou na kružnici a Delaunay triangulace není vytvořena správně. Pro grid byly vrstevnice znázorněny také kvalitně, přestože je z obrázku vidět, že Delaunay triangulace není vytvořena zcela správně a při novém generování je triangulace sama o sobě nejednoznačná (v případě, že jako první testujeme náhodný bod). Vrstevnice pro spočinek jsou také chybně vykresleny, jedna ze zásad vrstevnic je to, že se jedná o uzavřenou linii, vrstevnice se navzájem nesmí křížit a nesmí se dvojít. Bohužel z výstupu aplikace je patrné, že tuto chybu program nedokáže eliminovat. Generování vrstevnic sedla je absolutně špatně. Vrstevnice se kříží, dvojí se, zobrazení sedla, kde by

měla být ponechána mezera je vytvořena vrstevnice, která se na obou koncích duplikuje. Poměrně často je při generování sedla vytvořen i spočinek.

Pro sklon terénu je vyhodnocení také nejednoznačné. Pro hřbet i údolí je sklon vyobrazen korektně, neboť oblasti podél údolnice a hřbetnice mají shodnou výšku a oblasti směrem od hřbetnic/údolnic konstantně klesají/stoupají. Znázornění spočinku je také korektní, oblast je vykreslena bíle, tedy není zde žádný sklon, neboť oblast má konstantní výšku. Pro kupu je nekorektně vyobrazen střed, neboť v něm nejsou korektně vykresleny Delaunay trojúhelníky.

Vykreslení expozice je také nejednoznačné. Největší nedostatek je chybějící legenda, bez níž si uživatel není schopný představit, co která barva znamená. Pro zobrazení expozice kupy je pro vrchol opět špatně vypočtená. Ovšem pro následující místa je korektní a dle nastavených barev. Chybně je vyobrazena expozice i pro údolí a hřbet, kdy nastává stejný problém jako při gridu. Oblasti jsou vykresleny v závislosti na vytvořené triangulaci. Pro ukázkou je vložena i expozice kresby gridu, která je silně závislá na vytvořené triangulaci.

Co se týče importovaných dat, textové soubory neobsahují výrazné terénní útvary, které by mohly ovlivnit výpočty. Navíc jsou podrobné body zaměřovány v okolí výrazných terénních útvarů ve větším množství. Vrstevnice, sklon i expozice byly pro importovaná data vykresleny korektně v rámci lineární interpolace. Pro kvalitnější výsledek by bylo vhodné použít nějakou funkci pro vyhlazení vrstevnic či pro tvorbu vrstevnic prostřednictvím morfologické interpolace.

10 Náměty na vylepšení

10.1 Vrstevnice

V aplikaci by bylo vhodné zobrazit popis jednotlivých vrstevnic dle kartografických zásad, aby měl uživatel jasnou představu o nadmořské výšce dané oblasti. Z důvodu čitelnosti byly totiž odstraněny popisy bodů obsahující nadmořskou výšku. Dále by stálo za zmínku přidat možnost pro nastavení výpočtu vrstevnic v nějakém rozsahu.

10.2 Import dat

V rámci aplikace by mohla být možnost volby odstranění bodů, nad kterými nechce uživatel počítat triangulaci. Při geodetickém zaměřování jsou výsledkem měření z totální stanice i orientační body, které se mohou nalézat ve výrazné vzdálenosti od zaměřovaných podrobných bodů a které značně zneklidní vizualizaci. Pro použití této aplikace je tedy vhodné tyto body odstranit ručně.

10.3 Sklon terénu

Sklon terénu, jak je prostřednictvím aplikace vykreslován, vyvolává jen vizuální efekt. Uživatel si udělá přibližnou představu o terénu, ale tím to končí. Určitě by bylo vhodné sklonitost rozdělit do intervalů. V případě, že by aplikace měla praktické využití, kdy by jí například využívali stavaři, kteří při výkopu musí zkontrolovat sklon vzhledem k druhu horniny a přípustnému sklonu svahu, by bylo rozdělení barev dle intervalů určitě vhodné.

10.4 Tlačítka aplikace

Jedním z možných vylepšení aplikace je určitě nastavení dostupnosti tlačítek (Unabled, Disabled). V aplikaci je prozatím vyřešen jen neplatný vstup intervalu vrstevnic.

11 Reference

1. BAYER, Tomáš. Geometrické vyhledávání [online][cit. 1. 12.2018].
Dostupné z: <https://web.natur.cuni.cz/bayertom/images/courses/Adk/adk5.pdf>
2. JANEČKA, Karel, PACINA, Jan. Výukové materiály k předmětu KMA/UGI -
Západočeská univerzita v Plzni. [online][cit. 4. 12. 2018]
Dostupné z: <https://kgm.zcu.cz/studium/ugi/cviceni/ch08s01.html>
3. BENEŠ, Petr. Diplomová práce: Voroného diagramy v molekulární chemii. [online]
[cit. 4. 12. 2018]
Dostupné z: https://is.muni.cz/th/m1tcs/benes_dp.pdf
4. PUCKEY, Jonathan. Delaunay Raster. [online] [cit. 4. 12. 2018]
Dostupné z: <https://jonathanpuckey.com/projects/delaunay-raster/index.html>
5. VÚGTK. Terminologický slovník zeměměřictví a katastru nemovitostí. [online] [cit.
5. 12. 2018]
Dostupné z: <https://www.vugtk.cz/slovník/index.php>