

České vysoké učení technické v Praze
Fakulta stavební



Algoritmy v digitální kartografii

Digitální model terénu a jeho analýzy

Bc. Robin Pflug
Bc. Tomáš Klemsa

Obsah

1	Zadání úlohy	2
1.1	Bonusové úlohy	2
2	Obecná formulace a řešení problému	3
3	Aplikované algoritmy	4
3.1	Delaunay triangulace	4
3.1.1	Algoritmus DT Incremental	4
3.1.2	Algoritmus DT Incremental - add	5
3.2	Vrstevnice	6
3.2.1	Algoritmus tvorby vrstevnic	6
3.3	Sklon terénu	7
3.3.1	Algoritmus analýzy sklonu	7
3.4	Expozice terénu	8
3.4.1	Algoritmus analýzy expozice	8
4	Bonusové úlohy	9
4.1	Výběr barevných stupnic při vizualizaci sklonu a expozice	9
4.2	Barevná hypsometrie	9
5	Vstup dat do aplikace	10
6	Výstup aplikace	10
7	Dokumentace	11
7.1	Třídy	11
7.1.1	Algorithms	11
7.1.2	Edge	12
7.1.3	QPoint3D	12
7.1.4	SortbyX	12
7.1.5	Triangle	13
8	Testování	14
8.1	Test 1	15
8.2	Test2	17
8.3	Test3	19
9	Závěr	21
10	Náměty pro vylepšení	21
11	Reference	21

1 Zadání úlohy

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 DMT, 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 DMT z kartografického hlediska, zamyslete se nad slabinami algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

1.1 Bonusové úlohy

Ze zadání bonusových úloh byla řešena *Barevná hypsometrie* a *Výběr barevných stupnic při vizualizaci sklonu a expozice*.

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

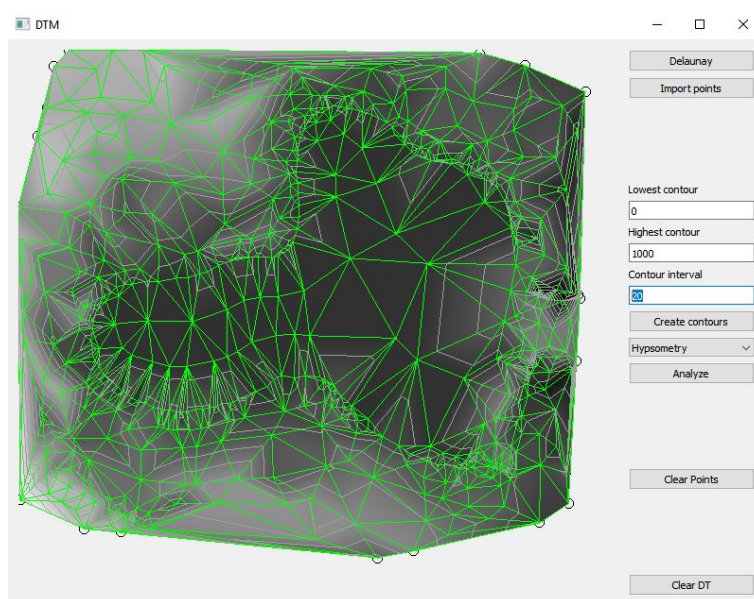
Obrázek 1: Bodové hodnocení úlohy [zdroj: 1]

2 Obecná formulace a řešení problému

Delaunayho triangulace: Při triangulaci množiny bodů, je žádoucí aby vytvořené trojúhelníky byly co nejvíce rovnostranné. Pokud to takto provedeme, každý trojúhelník by měl co nejlépe lokálně reprezentovat hodnotu povrchu. Další požadovaná charakteristika triangulačního procesu je, aby byla produkována jednoznačná triangulace nezávisle na počátečním bodě nebo orientaci množiny dat. Tyto výsledky budou předpověditelné a jednoduše opakovatelné. Sibson (1978) ukázal, že Delaunayho triangulace tyto podmínky obecně splňuje, i přes to, že jsou určité konfigurace množiny dat (jako např. pravouhlý GRID) která má na výstupu lokálně nejednoznačné řešení. [zdroj: 3]

Digitální model terénu: Digitální model terénu (DMT) je digitální popis a prezentace reálného povrchu jako 2D nebo 3D model, který se skládá z reálných naměřených dat a interpolačních metod, které dopočítávají pravděpodobná data pro místa, kde data chybí. Uvnitř modelovaného území je možno v libovolných bodech odvodit nadmořské výšky. [zdroj: 2]

Barevná hypsometrie: Barevná hypsometrie je kartografická technika znázornění terénního reliéfu na mapě, a to pomocí vrstevnic a plošného vybarvení jednotlivých výškových vrstev mezi nimi, tedy znázornění nadmořské výšky v mapách metodou vyplnění barevných ploch. [zdroj: 2]



Obrázek 2: Barevná hypsometrie

Zadáním této úlohy bylo vytvořit aplikaci, které nad zadanou množinou bodů vytvoří metodou inkrementální konstrukce 2D Delaunay triangulace. Takto vzniklá triangulace bude generovat digitální model terénu. V aplikaci dále budou dvě metody pro analýzu DTM: *sklon* a *expozici*. Nad DTM bude možné se zadaným krokem vygenerovat vrstavnice. Jako bonusová část úlohy je metoda, která vytvoří nad DTM barevnou hypsometrii.

3 Aplikované algoritmy

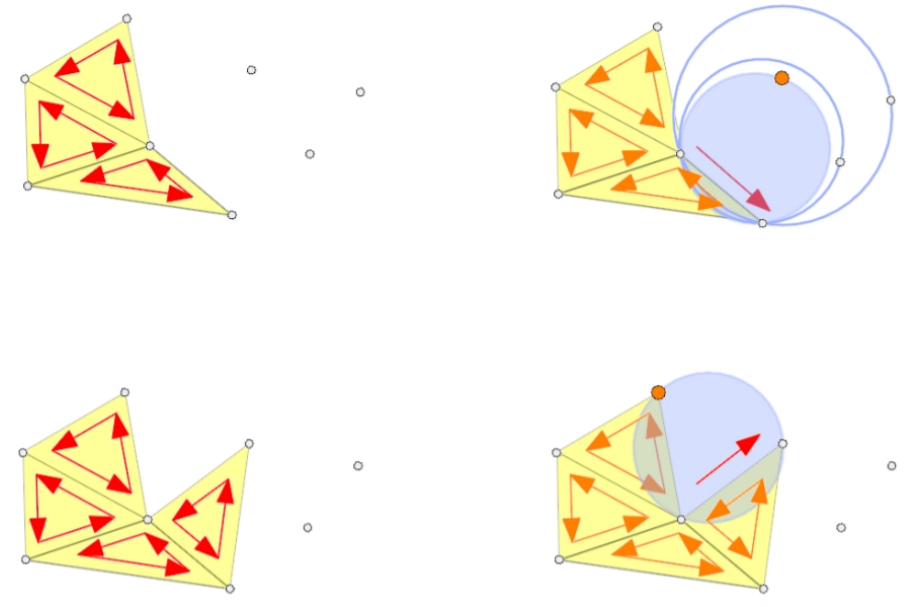
Algoritmy využívané pro tvorbu triangulace nad množinou bodů, digitálního modelu terénu, generaci vrstevnic a barevné hypsometrie.

3.1 Delaunay triangulace

Pro tvorbu triangulace byla použita Delaunay. Tato triangulace je nejběžnější pro metody tvořící digitální model terénu.

Algoritmus, který byl použit tvoří triangulaci *inkrementální konstrukcí*: Hrany mají danou orientaci a postupně jsou hledané takové body, které náležejí levé polorovině dané hranou a zároveň s body hrany tvoří minimální opsanou kružnici. Existuje-li takovýto bod, pak generuje dvě nové hrany, které jsou přidány do triangulace. Nepodaří-li se bod splňující tato kritéria nalézt, pak se orientace hrany otočí a bod je opět hledán.

Jednotlivé hrany jsou ukládány do listu aktivních hran (*AEL*). Pokud k hraně v *AEL* bude nalezen třetí bod, podle výše uvedených kritérií, bude hrana z listu vyjmuta. V *AEL* je také kontrolováno, zda se vkládaná hrana v listu již nenachází ovšem s opačnou orientací. Celý algoritmus probíhá, dokud list aktivních hran není prázdný.



Obrázek 3: Inkrementální konstrukce DT [zdroj: 1]

3.1.1 Algoritmus DT Incremental

1. Nalezení pivota $p_1 = rand(P)$, $||p_2 - p_1|| = min$
2. Vytvoř hranu $e = (p_1, p_2)$ / Nalezení prvních dvou bodů tvořících hranu.
3. $p_- = argmin_{\forall p_i \in \sigma_l(e)} r'(k_i)$, $k_i = (a, b, p_i)$, $e = (a, b)$ /

4. Pokud $\nexists p_-$, prohod' orientaci $e \leftarrow (p_2, p_1)$. Jdi na 3..
5. $e_2 = (p_2, p_-), e_3 = (p_-, p_1)$ // Zbývající hrany trojúhelníku
6. $AEI \leftarrow e, AEI \leftarrow e_2, AEI \leftarrow e_3$ // Přidání hran do listu aktivních hran
7. $DT \leftarrow e, DT \leftarrow e_2, DT \leftarrow e_3$ // Přidání hran do DT
8. while AEI not empty:
 - $AEI \rightarrow e, e = (p_1, p_2)$ // Vezmi první hranu z AEI
 - $e = (p_2, p_1)$ // Prohod' jejich orientaci
 - $p_- = \operatorname{argmin}_{\forall p_i \in \sigma_l(e)} r'(k_i), k_i = (a, b, p_i), e = (a, b)$
 - if $\exists p_-$: // Bod existuje
 - $e_2 = (p_2, p_-), e_3 = (p_-, p_1)$ // Zbývající hrany trojúhelníku
 - $DT \leftarrow e$ // Přidej hranu do DT ale ne do AEI
 - $\operatorname{add}(e_2, AEI, DT), \operatorname{add}(e_3, AEI, DT)$ // Přidej do DT i do AEI

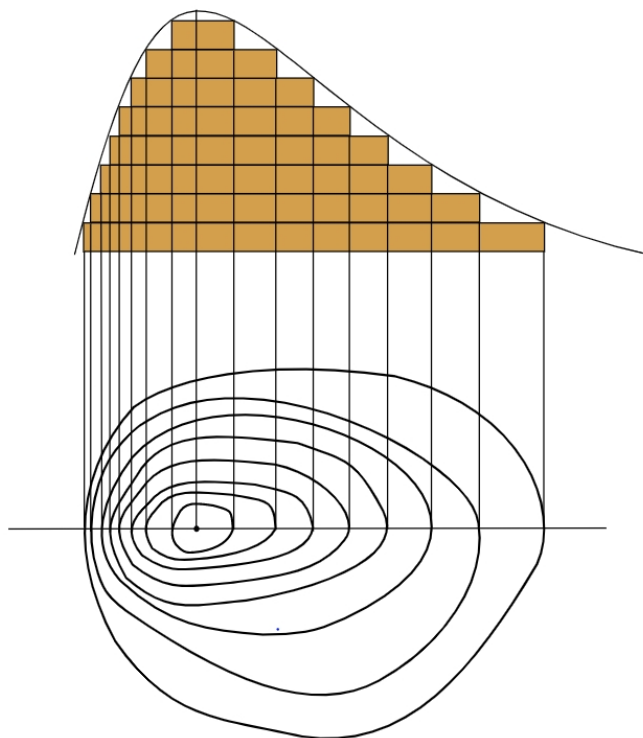
3.1.2 Algoritmus DT Incremental - add

1. Vytvoř hranu $e' = (b, a)$
2. if $(e' \in AEI)$
 - $AEI \rightarrow e'$
3. else:
 - $AEI \leftarrow e$
4. $DT \leftarrow (a, b)$

[zdroj: 1]

3.2 Vrstevnice

Vrstevnice je křivka, která v modelu spojuje body o stejné nadmořské výšce. Vrstevnice jsou generovány v pravidelném intervalu.



Obrázek 4: Grafické znázornění tvorby vrstevnic [zdroj: 2]

Vrstevnice byly v této úloze tvořeny lineární interpolací. Jako vrstevnice byla vygenerována přímka průsečíku vodorovné roviny o dané nadmořské výšce a příslušného trojúhelníku z modelu. Rozestup vrstevnic mezi dvěma body byl volen jako konstantní.

3.2.1 Algoritmus tvorby vrstevnic

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:

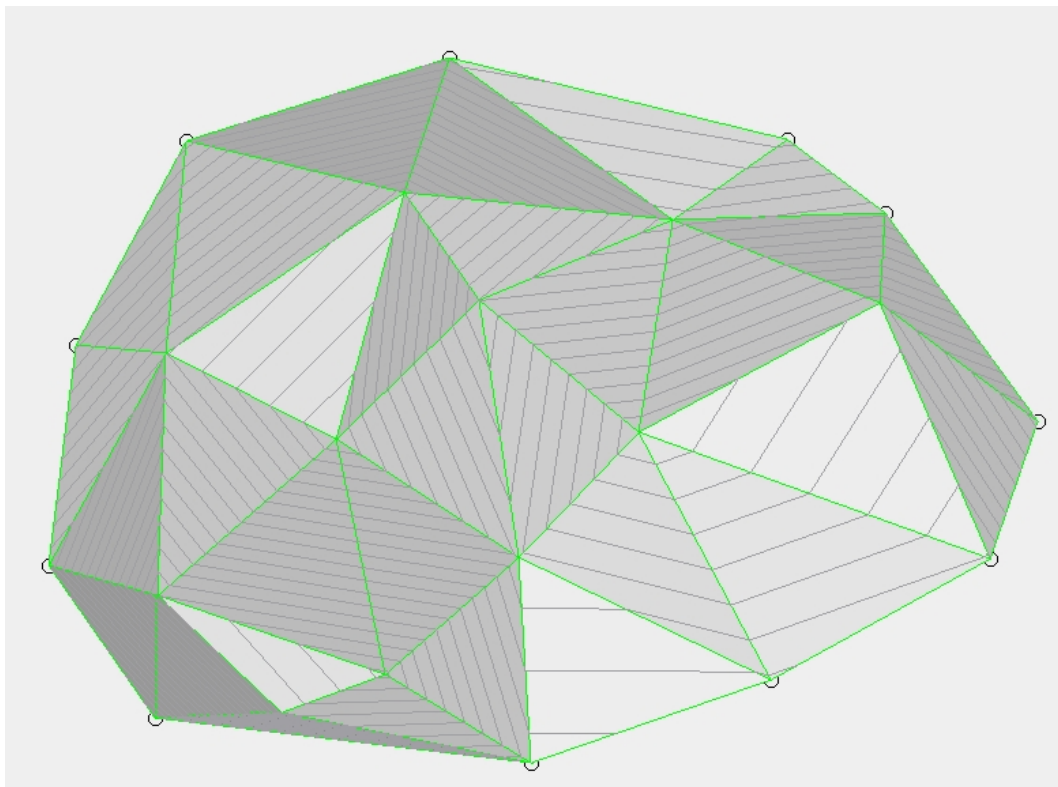
$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1$$

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

Vytvoř hranu tvořící vrstevnici.

3.3 Sklon terénu

Sklon terénu je určen jako úhel, který svírá normála roviny analyzovaného trojúhelníku a normála vodorovné roviny o velikosti 1. Sklon terénu byl graficky znázorněn stupněm odstínu šedé jednotlivých trojúhelníků. Tmavší odstín v analýze sklonu je důsledkem většího sklonu plošky.



Obrázek 5: Grafické znázornění sklonu

3.3.1 Algoritmus analýzy sklonu

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:

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

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

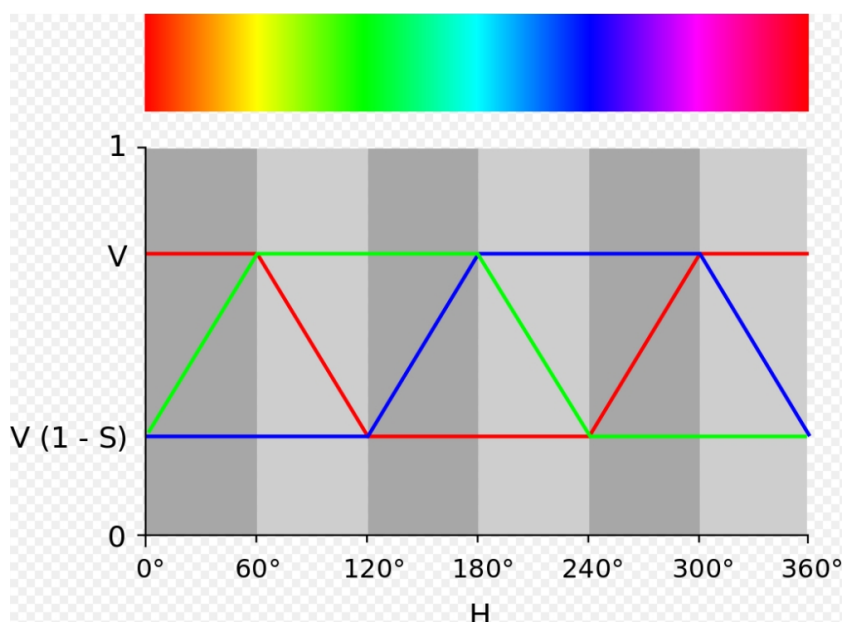
2. Výpočet sklonu: $\varphi = \arccos \frac{n_z}{|n_t|}$

3.4 Expozice terénu

Analýza expozice terénu je počítána jako azimut průmětu normálového vektoru trojúhelníku do rovny tvořené osami x a y . Expozice znázorňuje orientaci trojúhelníku tvořící model, vůči světovým stranám.

Pro vizualizaci expozice byl použit kulový barevný model HSV převedený do RGB. Tento barevný model nejvíce odpovídá lidskému vnímání barev. Skládá se ze tří složek (nejsou to základní barvy), u nichž je nutno hlídat hodnoty (možné nesmyslné kombinace):

- Hue – odstín
- Saturation – sytost barvy
- Value – hodnota jasu



Obrázek 6: Převod HSV modelu do RGB [zdroj: 2]

3.4.1 Algoritmus analýzy expozice

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: $A = \text{atan2}(\frac{n_x}{n_y})$

4 Bonusové úlohy

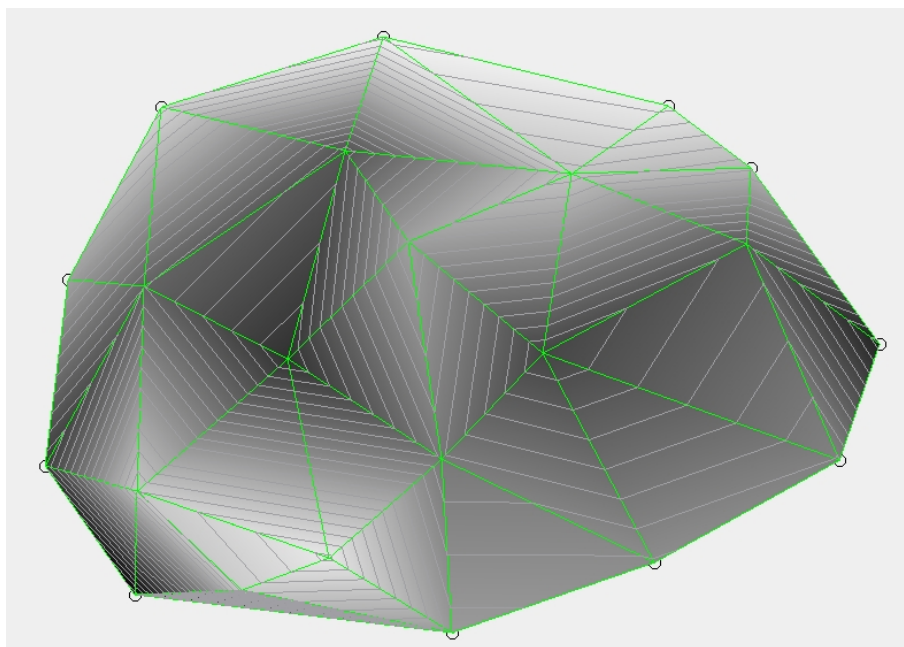
4.1 Výběr barevných stupnic při vizualizaci sklonu a expozice

Řešení této úlohy je popsáno v kapitole Aplikované algoritmy - *Sklon terénu a Expozice terénu*.

4.2 Barevná hypsometrie

Barevná hypsometrie spočívá ve vytvoření výškových vrstev ohraničených vrstevnicemi na rozhraní typických intervalů získaných z hypsografické křivky, a jejich vybarvení.

Barevná hypsometrie byla vytvořena na principu barevné škály "čím výše, tím světleji". Výhodou uvedeného postupu je stálost stupňování, které je přirozené pro kontinuitu povrchu terénu i přírodní výškové gradienty. [zdroj: 2]



Obrázek 7: Barevná hypsometrie aplikovaná na DTM

5 Vstup dat do aplikace

Vstup dat v aplikaci lze realizovat dvěma způsoby. Uživatel sám vkládá body do grafického okna za pomoci kurzoru myši nebo hromadně načte soubor ve formátu .txt obsahující strukturovaně uspořádané souřadnice bodů (viz vzor). Vkládaná množina bodů musí obsahovat zredukované souřadnice Z. Tato souřadnice musí být redukována na interval od 0 do 255. V případě, že hodnoty Z nepokrývají interval, je vhodné je přeskálovat na celý interval.

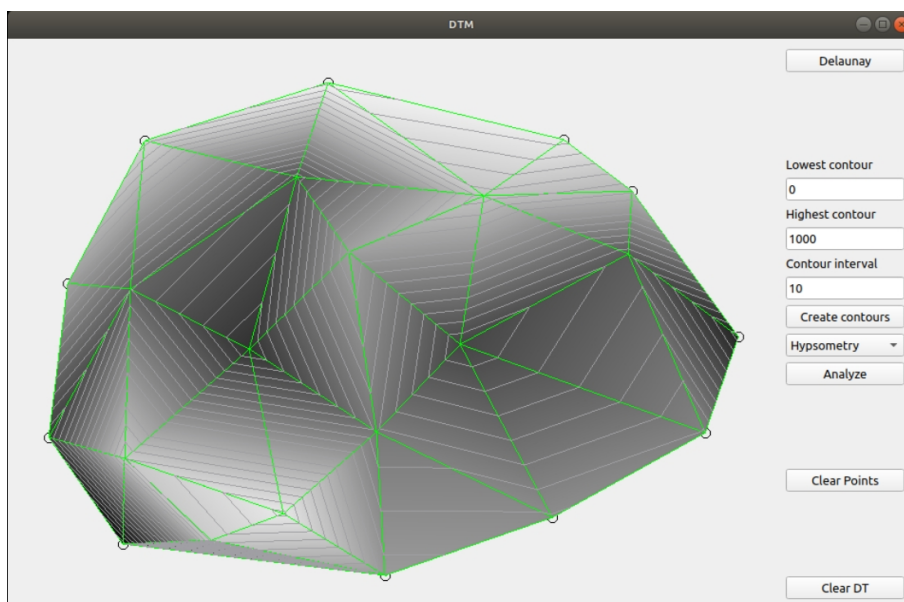
Grafický vstup souřadnic, tvořený kurzorem je náhodný. Proto je vhodný spíše pro testování programu.

737997.205	1055055.26	187.8
737917.947	1055095.562	188.65
737929.277	1055288.253	192.51

Obrázek 8: Vzor vstupních dat pro nahrávání ze souboru .txt

6 Výstup aplikace

Grafický výstup je prezentován v Canvasu grafického rozhraní.



Obrázek 9: Grafické okno aplikace

7 Dokumentace

7.1 Třídy

7.1.1 Algorithms

Třída Algorithms obsahuje metody zajišťující výpočty daných algoritmů.

int getPointLinePosition(QPoint3D q, QPoint3D p1, QPoint3D p2)

Návratová hodnota: *integer*;

Metoda vrátí polohu bodu *q* vůči přímce dané body *p1* a *p2*. Hodnota 1 je pro polohu vlevo od přímky, 0 pro polohu vpravo od přímky a -2 pro polohu bodu na přímce.

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

Návratová hodnota: *double*;

Metoda vrátí poloměr kružnice dané třemi body.

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

Návratová hodnota: *integer*;

Metoda navrací index nejbližšího bodu vzhledem k bodu *p* z vkládaného vektoru bodů *points*.

double distance2Points(QPoint3D p1, QPoint3D p2)

Návratová hodnota: *double*;

Metoda vrátí velikost vzdálenosti dvou bodů.

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

Návratová hodnota: *int*;

Metoda navrátí index ideálního Delaunay bodu ze zadaného vektoru bodů, vůči zadané úsečce.

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

Návratová hodnota: *vector< Edge >*;

Metoda navrací vektor hran tvořící Delaunayho triangulaci.

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

Návratová hodnota: *QPoint3D*;

Metoda vrací bod vrstevnice. Bod byl určen jako průsečík zadané přímky a vodorovné roviny o zadané výšce.

vector< Edge > createContourLines(vector< Edge > dt, double z_{min} , double z_{max} , double dz)

Návratová hodnota: *vector< Edge >*;

Metoda vytvoří a navrátí vektor obsahující hrany vrstevnice.

double calculateSlope(QPoint3D p1, QPoint3D p2, QPoint3D p3)

Návratová hodnota: *double*;

Metoda navrací hodnotu sklonu zadaného trojúhelníku.

double calculateAspect(QPoint3D p1, QPoint3D p2, QPoint3D p3)

Návratová hodnota: *double*;

Metoda navrací hodnotu azimutu zadaného trojúhelníku.

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

Návratová hodnota: *vector< Triangle >*;

Metoda navrací vektor analyzovaných trojúhelníků (tři body trojúhelníku, sklon a azimut) pro zadanou Delaunayho triangulaci.

7.1.2 Edge

Třída Edge definuje hranu. Hrana je dána jako počáteční bod *s* a koncový bod *e*.

QPoint3D getStart()

Návratová hodnota: *QPoint3D*;

Metoda vracející počáteční bod linie.

QPoint3D getEnd()

Návratová hodnota: *QPoint3D*;

Metoda vracející koncový bod linie.

void setStart(QPoint3D s)

Návratová hodnota: *void*;

Metoda nastaví počáteční bod linie.

void setEnd(QPoint3D e)

Návratová hodnota: *void*;

Metoda nastaví koncový bod linie.

7.1.3 QPoint3D

Třída QPoint3D definuje bod o třech souřadnicích (*x*; *y*; *z*).

double getZ()

Návratová hodnota: *double*;

Metoda vracející hodnotu souřadnice *z*.

void setZ(double z)

Návratová hodnota: *void*;

Metoda nastaví pro bod *QPoint3D* souřadnici *z*.

7.1.4 SortbyX

Třída SortbyX slouží k porovnání souřadnic v ose *x*.

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

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

7.1.5 Triangle

Třída Triangle definuje trojúhelník. Obsahuje tři vrcholy trojúhelníku *QPoint3D*, s vrcholy nese trojúhelník i hodnotu sklonu a azimutu.

QPoint3D getP1()

Návratová hodnota: *QPoint3D*;

Metoda vracející vrchol trojúhelníku P1.

QPoint3D getP2()

Návratová hodnota: *QPoint3D*;

Metoda vracející vrchol trojúhelníku P2.

QPoint3D getP3()

Návratová hodnota: *QPoint3D*;

Metoda vracející vrchol trojúhelníku P3.

double getSlope()

Návratová hodnota: *double*;

Metoda vracející hodnotu sklonu trojúhelníku.

double getAspect()

Návratová hodnota: *double*;

Metoda vracející hodnotu azimutu trojúhelníku.

void setP1(QPoint3D p1_)

Návratová hodnota: *void*;

Metoda nastaví trojúhelníku vrchol P1.

void setP2(QPoint3D p2_)

Návratová hodnota: *void*;

Metoda nastaví trojúhelníku vrchol P2.

void setP3(QPoint3D p3_)

Návratová hodnota: *void*;

Metoda nastaví trojúhelníku vrchol P3.

void setSlope(double slope_)

Návratová hodnota: *void*;

Metoda nastaví trojúhelníku sklon.

void setAspect(double aspect_)

Návratová hodnota: *void*;

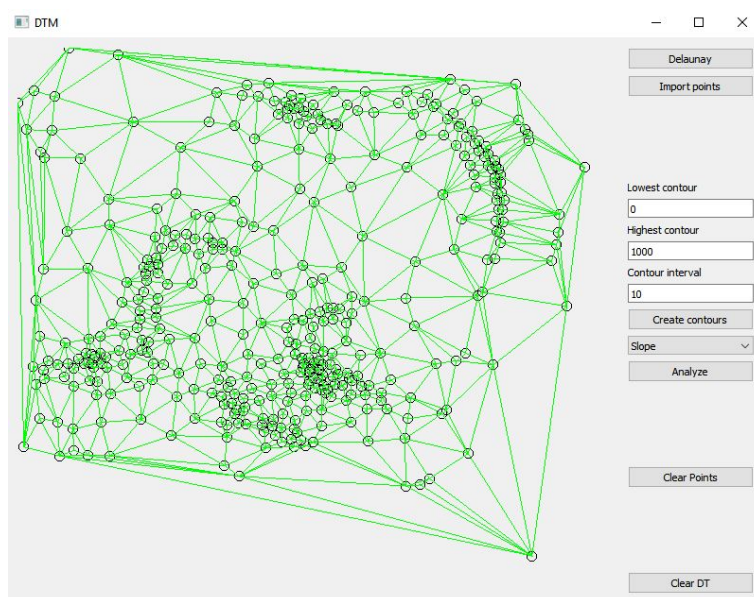
Metoda nastaví trojúhelníku azimut.

8 Testování

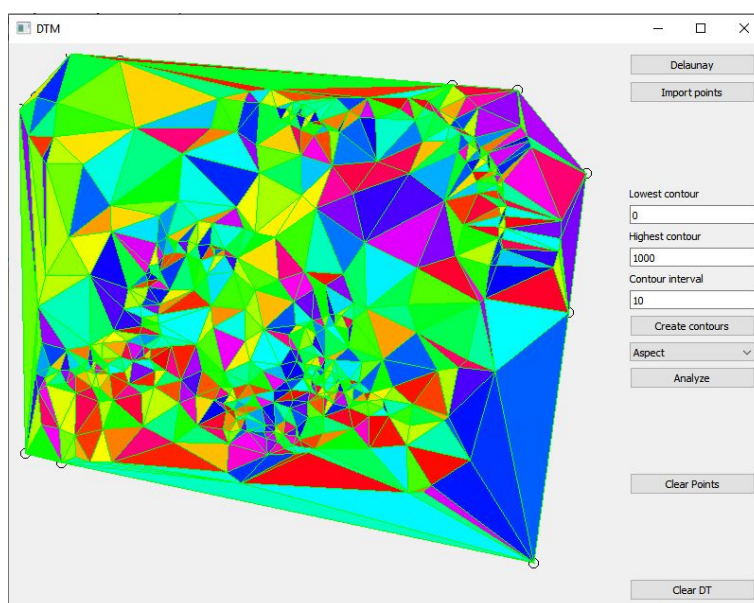
Testování probíhalo na reálných open-source datech poskytovaných na stránkách geoportálu Prahy: www.geoportalpraha.cz. Jediná úprava spočívala v redukci výškových souřadnic viz kapitola *Vstup dat do aplikace*. Z geoportálu byl stažen Digitální model terénu ve formátu shapefile. V softwaru QGis byly nad Digitálním modelem terénu části Prahy vygenerovány body v souřadicích S-JTSK a BpV. Vygenerovány byly 3 různé množiny bodů o rozdílné poloze a počtu.

8.1 Test 1

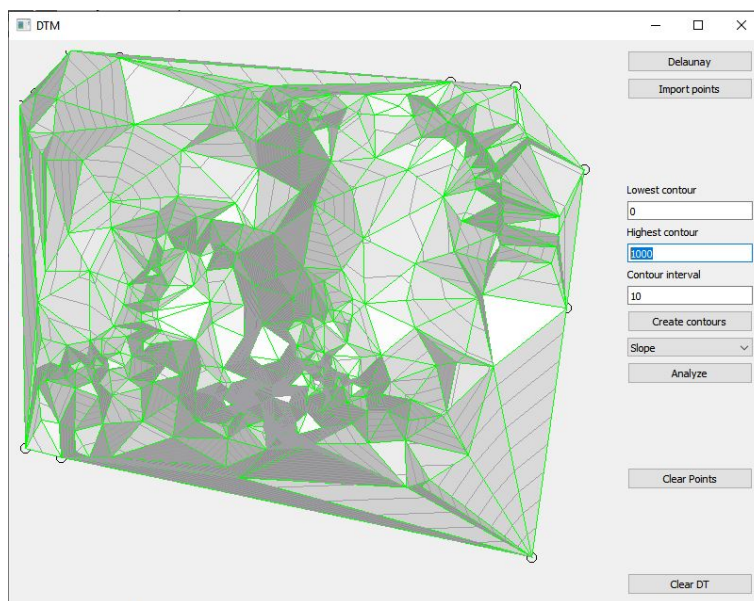
První množina použita pro testování dat obsahovala 389 bodů. V terénu se nacházely lesy, budovy, dopravní komunikace i pole.



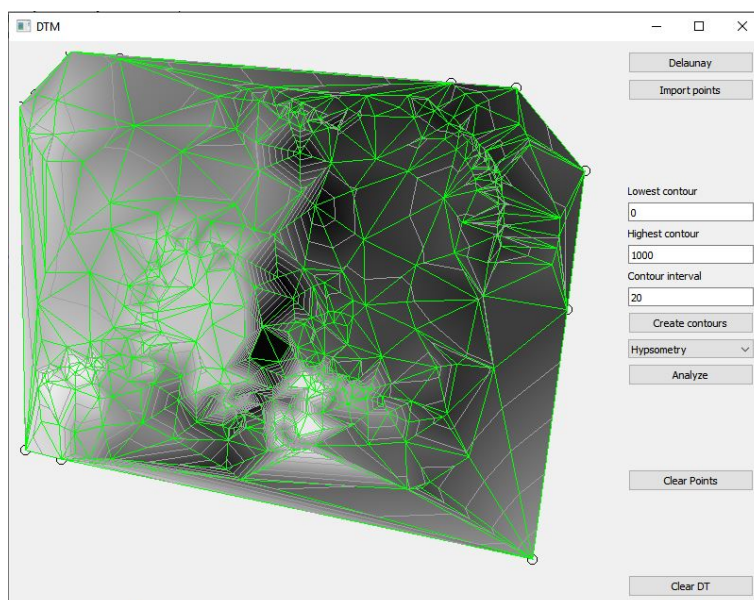
Obrázek 10: Delaunay triangulace



Obrázek 11: Analýza azimutu



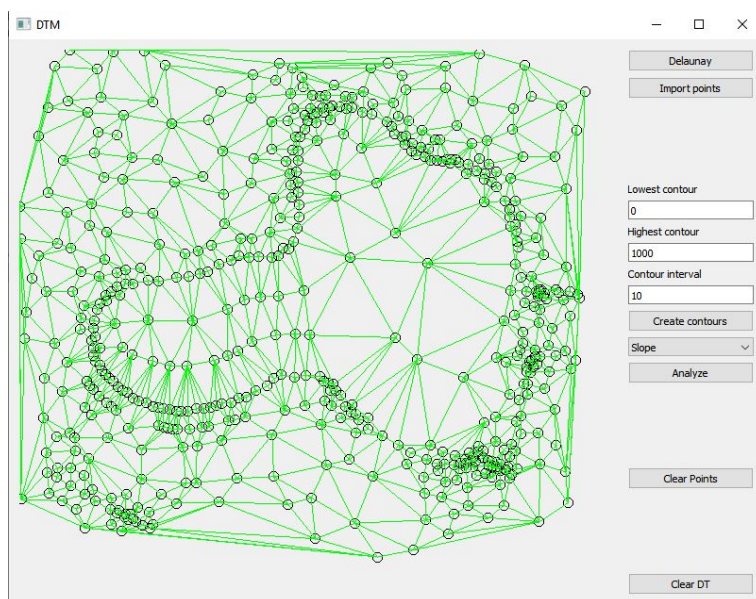
Obrázek 12: Analýza sklonu



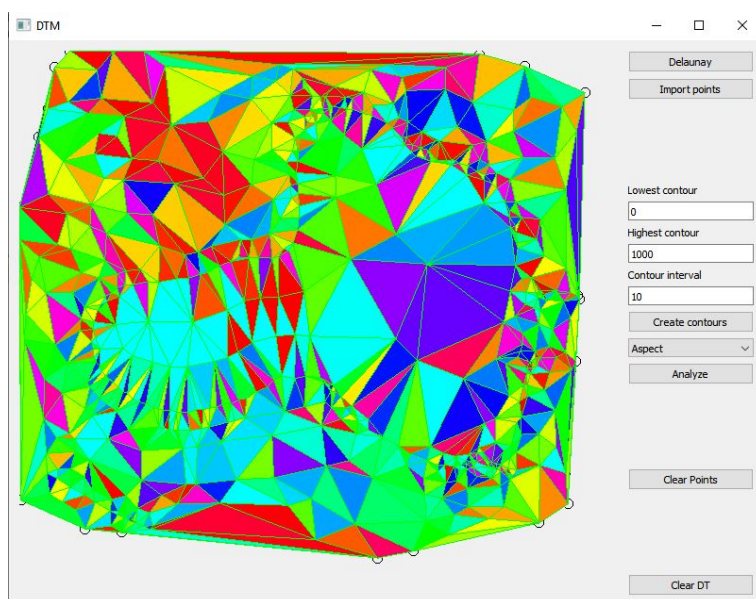
Obrázek 13: Barevná hypsometrie

8.2 Test2

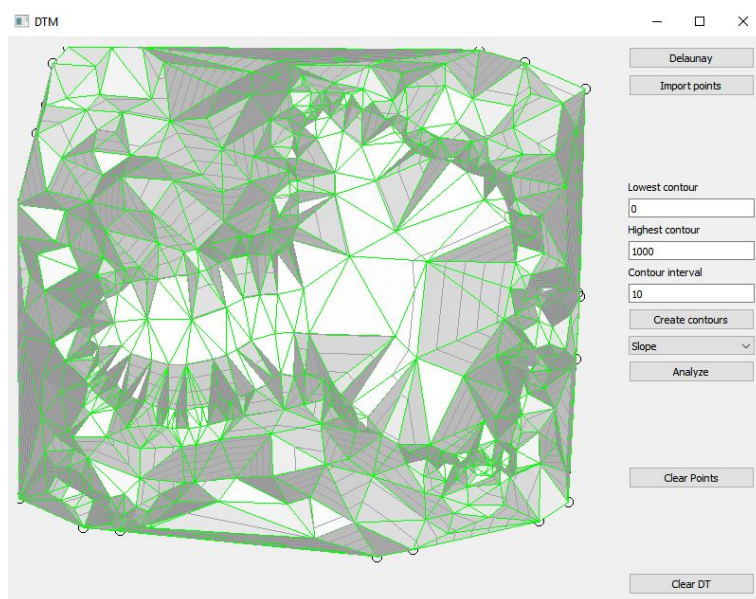
Druhá množina použita pro testování dat obsahovala 527 bodů. V terénu se nacházel rybník.



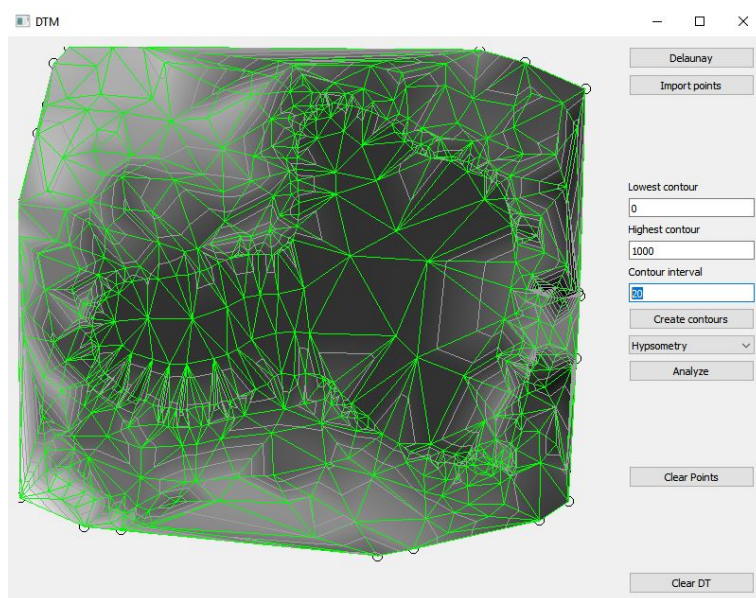
Obrázek 14: Delaunay triangulace



Obrázek 15: Analýza azimutu



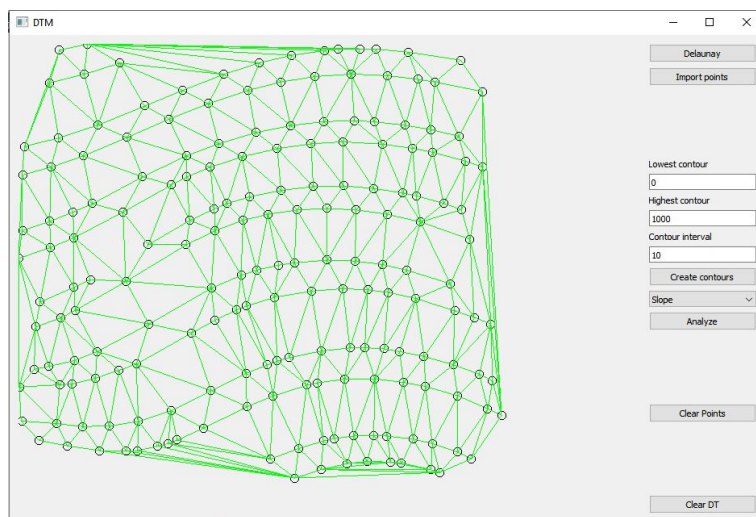
Obrázek 16: Analýza sklonu



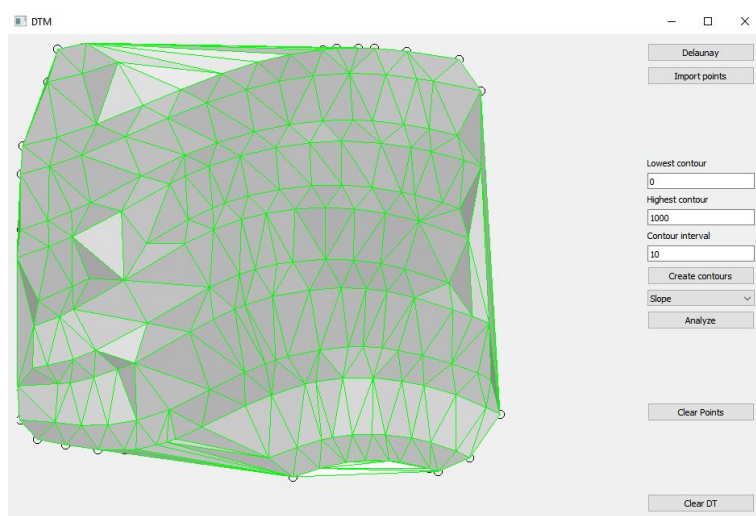
Obrázek 17: Barevná hypsometrie

8.3 Test3

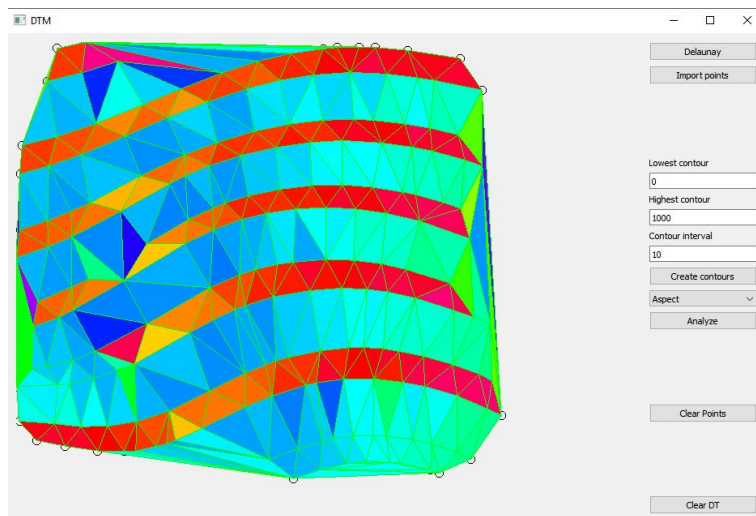
Třetí množina použita pro testování dat obsahovala 187 bodů. V terénu se nacházelo pole. Z aplikace lze pozorovat přesnot využitého modelu terénu. V důsledku přeškálování rozsahu souřadnice Z jsou zde vidět jednotlivé nerovnosti terénu.



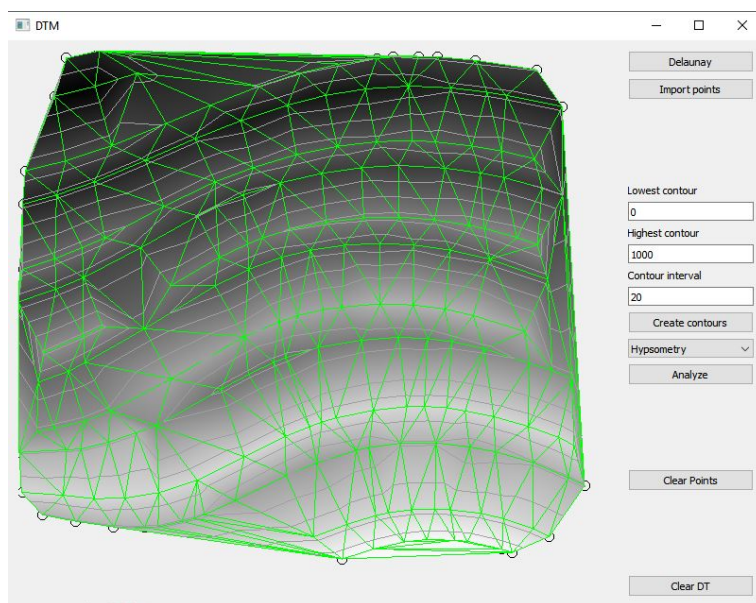
Obrázek 18: Delaunay triangulace



Obrázek 19: Analýza azimutu



Obrázek 20: Analýza sklonu



Obrázek 21: Barevná hypsometrie

9 Závěr

Aplikace vytváří a analyzuje digitální model povrchu vytvořený Delaunay triangulací. Analyzován je sklon a azimut terénu (trojúhelníků). Aplikace dále vykresluje vrstevnice v zadaném intervalu a lze v ní generovat barevnou hypsometrii.

Delaunay triangulace pro území s nevýraznými terénními anomáliemi funguje přijatelně. Problém zde ovšem nastává v případech, kdy neexistuje jednoznačné řešení (body na mřížce apod.).

Analýza sklonu i azimutu terénu selhává pro rovinná území (dále např. vodní plochy apod.). Tento nedostatek je způsoben zaokrouhlením, kde takto způsobené chyby zkreslují výsledek. Uživatel pak může nabýt nejistoty, zda je terén opravdu rovinný.

Tvorba vrstevnic je v aplikaci realizována lineární interpolací, ve skutečnosti však vrstevnice mají kontinuální průběh křivek. Pro lepší přiblížení se skutečným vrstevnicím, by bylo pro téměř veškeré terénní útvary vhodnější použít morfologickou interpolaci.

Do aplikace lze nahravát i větší množiny bodů v tetovém formátu bez potřeby větších úprav. Proto lze aplikaci využít na modelaci a následnou analýzu naměřených hodnot rovinných území například metodou GNSS.

10 Náměty pro vylepšení

Aplikace, konkrétně Delaunay triangulace, obecně selhává u bodů vytvářejících strukturu mřížky. Dále také selhává u určitých terénních tvarů (hrana, propast). Jako řešení těchto problémů se využívá povinných hran, které ovšem v aplikaci definovat nelze.

Dalším zlepšení vyplývajícím ze předchozího odstavce je naopak vymazání existujících hran, které nedávají smysl.

Pro vrstevnice by také bylo vhodnější generovat popis se skutečnými výškami v daném souřadnicovém systému.

11 Reference

1. BAYER, Tomáš. Metody konstrukce konvexní obálky [online][cit. 5.11.2019]. Dostupné z: <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf>
2. WIKIPEDIE Otevřená encyklopedie [online][cit. 2.12.2019]. Dostupné z: cs.wikipedia.org
3. Katedra geomatiky - Západočeská univerzita, Delaunayho triangulace [online][cit. 2.12.2019].

Dostupné z: <https://kgm.zcu.cz/studium/ugi/cviceni/ch08s01.html>