Úloha 3 – Digitální model terénu

Algoritmy počítačové kartografie

Sabina Fukalová, Klára Linková

Praha 2023

1. Zadání úlohy

1.1. Povinná část

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. 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 3 strany formátu A4.

1.2. Volitelná část

Barevná hypsometrie 5 b.

2. Popis a rozbor problému

Cílem triangulace T nad množinou bodů P = {P_i} je vytvořit soubor m trojúhelníků t = {t_1. t_2,…,t_m} a hran tak, aby byly splněny následující podmínky:

- jakékoli dva různé trojúhelníky mají společnou nejvýše jednu hranu,
- sjednocení všech trojúhelníků vytvoří v E^2 souvislou množinu,
- uvnitř generovaných trojúhelníků neleží žádný další bod z P (Bayer 2008).

2.1. Delaunay triangulace

Delaunayova triangulace (DT) je nejčastěji používanou triangulací (Bayer 2023). Jedná se o množinu m trojúhelníků {t_1. t_2,…,t_m} a hran vytvořených nad množinou vstupních bodů P tak, aby platilo:

- každý bod množiny P je vrcholem alespoň jednoho trojúhelníku t_i,
- dva libovolné různé trojúhelníky mají společnou nejvýše jednu hranu,
- pokud nad každým trojúhelníkem vytvoříme opsanou kružnici procházející vrcholy V_1, V_2, V_3, tak žádný další bod neleží uvnitř této kružnice,
- DT maximalizuje minimální úhel,
- DT neminimalizuje maximální úhel,
- DT je lokálně optimální a globálně optimální (Bayer 2008).

DT je jednoznačná pouze v případě, neleží-li žádné čtyři body na kružnici. Pokud není tato podmínka splněna, existuje více variant DT. Výsledkem triangulace je konvexní obálka.

2.1.1. Matematický popis DT

Následující text vychází z Bayer (2008). Vycházíme z obecné rovnice kružnice se středem S = [m, n], která prochází třemi body $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$. Dostaneme rovnice

$$
\begin{aligned}
(x_1 - m)^2 + (y_1 - n)^2 &= r^2 \\
(x_2 - m)^2 + (y_2 - n)^2 &= r^2 \\
(x_3 - m)^2 + (y_3 - n)^2 &= r^2.
\end{aligned}
$$

Neznámé *m*, *n* se určí pomocí substituce

$$
\begin{aligned}
A &= x_2^2 - x_1^2 \qquad & B &= 2x_1 - 2x_2, \\
C &= y_2^2 - y_1^2 \qquad & D &= 2y_1 - 2y_2, \\
E &= x_3^2 - x_1^2 \qquad & F &= 2x_1 - 2x_3, \\
G &= y_3^2 - y_1^2 \qquad & H &= 2y_1 - 2y_3,
\end{aligned}
$$

Po dosazení

$$m = \frac{D(E+G) - H(A+C)}{BH - FD},$$

$$n = \frac{-A - mB - C}{D},$$

za předpokladu, že BH ≠ FD a D ≠ 0.

Test vzájemné polohy bodu a kružnice lze aplikovat různými způsoby. Prvním z nich je dosazením souřadnic středu kružnice m, n a poloměru r do rovnice

$$(x-m)^2 + (y-n)^2 - r^2 = p.$$

V závislosti na hodnotě p mohou nastat tři případy polohy:

- $p < 0$ …bod leží uvnitř kružnice,
- $p = 0$ …bod leží na kružnici,
- $p > 0$ …bod leží vně kružnice.

Další metodou pro určení polohy bodu a kružnice je za pomoci determinantu. Na vstupu je uspořádaná čtveřice bodů P_1, P_2, P_3, P_4, přičemž bod P_1 představuje testovaný bod a body P_2, P_3, P_4 leží na kružnici. Nejprve se testuje, zda má trojice bodu kladnou orientaci (ve směru hodinových ručiček), tedy platím že determinant

$$\begin{vmatrix} P_1(x) & P_1(y) & 1 \\ P_2(x) & P_2(y) & 1 \\ P_3(x) & P_3(y) & 1 \end{vmatrix} > 0.$$

Je-li podmínka splněna, testuje se hodnota determinantu

$$\begin{vmatrix} P_1(x) & P_1(y) & P_1(x)^2 + P_1(y)^2 \\ P_2(x) & P_2(y) & P_2(x)^2 + P_2(y)^2 \\ P_3(x) & P_3(y) & P_3(x)^2 + P_3(y)^2 \\ P_4(x) & P_4(y) & P_4(x)^2 + P_4(y)^2 \end{vmatrix}.$$

Je-li D < 0, leží bod uvnitř opsané kružnice, je-li D > 0, bod leží vně opsané kružnice. Pokud D = 0, leží bod na kružnici.

Třetím možným testem je tzv. Cline-Renka test, který ověřuje legalitu dvojice trojúhelníků sdílejících společnou stranu s vrcholy P_1, P_2, P_3, P_4. Pokud označíme vrcholové úhly protilehlé ke společné straně jako α, β, pak mohou nastat tři situace

- $\alpha + \beta < \pi$…triangulace je legální,
- $\alpha + \beta = \pi$…body P_1, P_2, P_3, P_4 leží na kružnici,
- $\alpha + \beta > \pi$…triangulace je nelegální a je třeba prohodit hrany ve čtyřúhelníku.

K prohození hrany tedy dojde v případě

$$\sin(\alpha + \beta) = \cos\alpha \sin\beta + \cos\beta \sin\alpha < 0.$$

Tuto podmínku lze přepsat do tvaru

$$\frac{(x_{13}x_{23} + y_{13}y_{23})(x_{24}y_{14} - x_{14}y_{24}) + (x_{13}y_{23} - x_{23}y_{13})(x_{24}x_{14} + y_{24}y_{14})}{\sqrt{(x_{13}^2 + y_{13}^2)(x_{23}^2 + y_{23}^2)(x_{24}^2 + y_{24}^2)(x_{14}^2 + y_{14}^2)}} < 0,$$

kde

$$\begin{aligned} x_{13} &= x_3 - x_1 \qquad & y_{13} &= y_3 - y_1 \\ x_{23} &= x_2 - x_3 \qquad & y_{23} &= y_2 - y_3 \\ x_{14} &= x_1 - x_4 \qquad & y_{14} &= y_1 - y_4 \\ x_{24} &= x_2 - x_4 \qquad & y_{24} &= y_2 - y_4 \end{aligned}$$

Jmenovatel bude vždy kladný, proto lze testovací kritérium napsat následovně

$$(x_{13}x_{23} + y_{13}y_{23})(x_{24}y_{14} - x_{14}y_{24}) < (x_{23}y_{13} - x_{13}y_{23})(x_{24}y_{14} + y_{24}y_{14}).$$

2.1.2. Metoda inkrementálního vkládání

Princip této metody konstrukce Delaunayho triangulace spočívá v postupném přidávání bodů do již vytvořené DT. Algoritmus lze použít ve 2D i 3D. V prvním případě se pracuje s prázdnou kružnicí, ve druhém s prázdnou koulí. Složitost metody je $O(N^2)$. Metoda pracuje v následujících krocích (Bayer 2008):

1. Vybere se náhodný bod (pivot), který bude vrcholem některého z budoucích trojúhelníků, případně se zvolí bod s minimální/maximální souřadnicí x nebo y.
2. Najde se nejbližší bod k pivotu a vytvoří se hrana DT.
3. Postupně se prochází body vstupní množiny P a hledá se bod, pro který platí, že kružnice opsaná hraně a tomuto bodu má minimální poloměr. Souřadnice středu kružnice se spočítají z rovnic kružnice, při známých hodnotách m, n a poloměr z (viz kapitola 2.1.1). Nalezený bod je vrchol trojúhelníka DT.

4. Rekurzí je hledán ke každé nově vzniklé hraně takový bod, aby kružnice opsaná hraně a tomuto bodu měla minimální poloměr. Tento krok probíhá, dokud existuje nějaká neprohledaná hrana trojúhelníka.

Jiný způsob provedení této metody je s využitím Dealunayovy vzdálenosti *d(h,P)*, kdy se hledá bod P vzhledem k hraně h takový, aby platilo

$$d(h, P) = \min\big(d(h, P_i)\big),$$

kde P představuje bod s nejmenší Delunayovou vzdáleností a leží vpravo od hrany. Když takový bod nebyl nalezen, změní se orientace hrany a pokračuje se v hledání směrem vlevo od hrany. Takto vzniklý první trojúhelník je přidán do seznamu AEL (tzv. *Active Edge List*), který obsahuje hrany DT orientované v jednom směru. Pro tyto hrany jsou postupně hledány body P s nejmenší Delaunayovou vzdáleností. Pokud je nalezen takový trojúhelník a jeho dvě hrany P, P_1 a P, P_2 nejsou v AEL, jsou tyto hrany přidány do AEL (případně může být přidána pouze jedna hrana, pokud se dosud nenachází v AEL). Hrana, který se již nachází v AEL je přidána do DT. Pokud žádná z hran P, P_1 a P, P_2 není v AEL, je do DT přidána hrana P_1, P_2. Pokud pro hranu P_1, P_2 neexistuje žádný bod P, znamená to, že hrana leží na konvexní obálce a tato hrana je přidána do DT. Tento postup probíhá, dokud AEL není prázdná (Bayer 2008).

2.2. Polyedrický digitální model terénu

Polyedrický (trojúhelníkový) model terénu je tvořen ploškami ve tvaru nepravidelných trojúhelníků. Tyto trojúhelníky mají společnou nejvýše jednu hranu a pokrývají celé zájmové území. Síť trojúhelníků je vytvořena pomocí triangulačních algoritmů (nejčastěji 2D varianta DT). Pokud jsou vrcholy jednotlivých trojúhelníků proloženy roviny v R^3, vznikne nepravidelný mnohostěn (tzv. polyedr) přimykající se k terénu (Bayer 2008). Do polyedrického modelu lze přidávat povinné hrany (např. údolnice, hřbetnice, spádnice), čímž se zlepší aproximační vlastnosti modelu (Bayer 2023).

2.2.1. Konstrukce polyedrického modelu

Na vstupní data je aplikován triangulační algoritmus. Nad vzniklou trojúhelníkovou sítí je vytvořen polyedrický model tak, že vrcholy $P_1 = [x_1, y_1, z_1]$, $P_2 = [x_2, y_2, z_2]$, $P_3 = [x_3, y_3, z_3]$ každého trojúhelníku je proložena rovina

$$z = ax + by + c,$$

kde koeficienty a, b, c představují složky normálového vektoru roviny

$$a = \frac{\begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad b = \frac{\begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad c = \frac{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}.$$

2.2.2. Analýzy nad polyedrickým modelem

Polyedrický model je nejčastěji využíván pro přibližný výpočet průběhu vrstevnic, konstrukci profilů či řezů a další analýzy, protože výpočty nad polyedrickým digitálním modelem terénu jsou poměrně jednoduché (Bayer 2008).

Konstrukce vrstevnic

Vrstevnice je možné konstruovat metodou lineární interpolace, která je založena na hledání průsečnice roviny τ určené trojúhelníkem $t \in DT$ a vodorovné roviny s výškou h (Bayer 2023).

Vstupními parametry je výška vrstevnice z_0, kterou budeme konstruovat, a vrcholy trojúhelníku polyedrického modelu $V_1 = [x_1, y_1, z_1]$, $V_2 = [x_2, y_2, z_2]$, $V_3 = [x_3, y_3, z_3]$ definující rovinu terénu. Úkolem algoritmu je najít souřadnice bodů $P_1 = [x_{p1}, y_{p1}]$, $P_2 = [x_{p2}, y_{p2}]$, jejichž spojnice představuje průsečnici roviny $z = z_0$ s rovinou trojúhelníku V_1, V_2, V_3 (Bayer 2008).

Průsečíky P_1 a P_2 však nemusí vždy existovat, záleží na vzájemné poloze vrcholů vzhledem k rovině $z = z_0$. Pro tvorbu vrstevnic nás zajímají pouze situace, kdy

- dva vrcholy leží výše než vrstevnice, jeden níže,
- dva vrcholy leží níže než vrstevnice, jeden výše,
- jeden vrchol leží níže než vrstevnice, jeden výše než vrstevnice, třetí na vrstevnici.

Je testováno

$$\begin{aligned}
&(z - z_1)(z - z_2) > 0 \quad (z - z_1)(z - z_2) < 0 \quad (z - z_1)(z - z_2) = 0, \\
&(z - z_1)(z - z_3) > 0 \quad (z - z_1)(z - z_3) < 0 \quad (z - z_1)(z - z_3) = 0, \\
&(z - z_2)(z - z_3) > 0 \quad (z - z_2)(z - z_3) < 0 \quad (z - z_2)(z - z_3) = 0.
\end{aligned}$$

Jestli některý ze součinů je

- < 0…průsečík leží ve straně trojúhelníku,
- $= 0$…průsečík leží ve vrcholu trojúhelníku,
- > 0…průsečík neexistuje.

Souřadnice bodů P_1, P_2 průsečnice jsou určeny ze vztahů

$$x_{p1} = \frac{x_3 - x_1}{z_3 - z_1}(z_0 - z_1) + x_1 \qquad x_{p2} = \frac{x_2 - x_1}{z_2 - z_1}(z_0 - z_1) + x_1$$

$$y_{p1} = \frac{y_3 - y_1}{z_3 - z_1}(z_0 - z_1) + y_1 \qquad y_{p2} = \frac{y_2 - y_1}{z_2 - z_1} + (z_0 - z_1) + y_1.$$

Analýza sklonu

Sklon (angl. *slope*) je veličina, která udává velikost úhlu mezi normálou trojúhelníku a svislicí. Vzorec pro výpočet sklonu má tvar

$$\varphi = \arccos \frac{n_1 \cdot n_2}{\|n_1\| \cdot \|n_2\|} = arccos \frac{c}{\|n_1\|},$$

kde $n_1 = (a, b, c)$ je normálový vektor trojúhelníku, $n_2 = (0, 0, 1)$ je svislý vektor (Bayer 2023).

Analýza expozice

Orientace svahu neboli expozice (angl. *aspect*), popisuje orientaci povrchu vůči světovým stranám. Jedná se o azimut průmětu gradientu daného trojúhelníku do roviny xy. Vyjadřuje se ve stupních v rozsahu 0°–360°. Výpočet je proveden podle vztahu

$$A = \arctan \frac{a}{b}.$$

Barevná hypsometrie

Barevná hypsometrie je technika pro znázorňování výškopisu. Princip spočívá v plošném vybarvení jednotlivých výškových vrstev mezi vrstevnicemi podle vhodně zvolené barevné stupnice.

3. Dokumentace

Vytvořený program je rozdělen do sedmi modulů – *mainform.py, input.py, draw.py, algorithms.py, qpoint3d.py, edge.py, settings.py*.

3.1. Modul Mainform.py

Tento modul obsahuje třídu *Ui_MainForm*, která byla vytvořena automaticky pomocí Qt Creatoru. Dále jsou doplněny metody, které se spustí při interakci s uživatelským rozhraním. Metoda *openFile* volá funkci *loadFile* z modulu input.py, čímž jsou vstupní body ve formátu *.shp* nahrány do proměnné *points* ve třídě *Draw*. Metoda *runDT* volá algoritmus pro tvorbu Delaunayho triangulace z modulu *Algorithms.py*. Výsledná triangulace je uložena do listu v podobě jednotlivých hran ve třídě *Draw*. Metoda *runContourLines* vytvoří vrstevnice na základě Delaunayho triangulace. Parametry vrstevnic jsou načteny z uživatelského rozhraní pomocí modulu *settings.py*. Vrstevnice jsou uloženy jako list hran ve třídě *Draw*. Metoda computeSlope počítá sklon jednotlivých trojúhelníků Delaunayho triangulace a uloží ho do listu ve třídě *Draw*. Metoda *computeAspect* počítá orientaci jednotlivých trojúhelníků Delaunayho triangulace a ukládá ji do listu ve třídě *Draw*. Metoda *createColorH* počítá průměrnou nadmořskou výšku jednotlivých trojúhelníků Delaunayho triangulace a ukládá ji do listu ve třídě *Draw*. Dále jsou vytvořeny metody *clearAll, clearCL, clearSlope, clearAspect, clearColorH*, které mažou listy z daty ve třídě *Draw*. Poslední metodou je *showSettings*, která vyvolá dialogové okno uživatelského rozhraní pro nastavení parametrů při tvorbě vrstevnic.

3.2. Modul input.py

Modul *input.py* obsahuje metodu *loadFile* pro načtení vstupních dat. V této metodě je nejprve načtena cesta ke vstupním datům. Pokud není zvolena žádná cesta, dojde k ukončení metody. Dále jsou načtena vstupní data jako shapefile a převedena do formátu *QPoint3D*, díky čemuž obsahuje i informaci o nadmořské výšce bodu. Nadmořská výška je získána z atributů bodů, kde je uvedena ve čtvrtém sloupečku atributové tabulky. Tato data jsou transformována do lokálního souřadnicového systému, aby se zobrazila v definovaném okně uživatelského rozhraní. Transformovaná data jsou uložena jako jednotlivé body do listu *points*. V závěru metoda vrací list *points* a *scale* (informace o změně velikosti os *x* a *y*).

3.3. Modul draw.py

Modul *draw.py* obsahuje třídu *Draw*, kterou dědí od třídy *QWidget* z knihovny PyQt6. Díky této funkci jsou vykreslovány vstupní body a výsledky jednotlivých funkcí. Vstupní body jsou uloženy v listu *points*, výsledky funkcí v listech *dt* (Delaunayho triangulace), *cont_lines* (vrstevnice), *heights* (nadmořská výška trojúhelníků), *slope* (sklon trojúhelníků), *aspect* (orientace trojúhelníků). Vykreslení je definováno v metodě *paintEvent*. Dále třída *Draw* obsahuje *gettery* a *settery* pro editaci dat z ostatních modulů. Na závěr je vykreslení barevné hypsometrie, které je realizováno tak, že jsou nejprve vytvořeny polygony trojúhelníků a pro ně jsou určeny podmínky, díky nimž jsou trojúhelníky vybarveny na základě výšky – barva je nastavena ve formě kódu RGB.

3.4. Modul algorithms.py

V modulu *algorithms.py* se nachází jednotlivé algoritmy pro tvorbu digitálního modelu terénu. Je zde jedna třída *Algorithms*, která obsahuje metody pro tvorbu digitálního modelu terénu.

Metoda *getPointAndLineposition* má na vstupu 3 body QPoint3D – 1 analyzovaný bod a 2 body definující přímku. Tato metoda zjišťuje, ve které polorovině se analyzovaný bod nachází na základě determinantu. Pokud bod leží v levé polorovině, metoda vrací hodnotu 1. Pokud se bod nachází v pravé rovině, vrací hodnotu 0. V případě, že analyzovaný bod je kolineární, vrací metoda hodnotu -1.

Metoda *get2LinesAngle* má na vstupu 4 body QPoint3D, přičemž první 2 definují jednu úsečku a druhé 2 definují druhou úsečku. Nejprve jsou spočítány směrové vektory obou úseček a z nich je spočítán skalární součin a norma. Tyto hodnoty jsou poté dosazeny do vzorce pro výpočet úhlu mezi dvěma úsečkami.

Metoda *getCircleCenterAndRadius* má na vstupu 3 body QPoint3D, které definují kružnici. Z těchto bodů jsou vypočítány souřadnice středu kružnice a její poloměr. Střed kružnice je vracen jako QPoint3D, poloměr jako *float*.

Metoda *getNearestPointIdx* má na vstupu bod QPoint3D a list *points* všech bodů QPoint3D. Metoda prochází všechny body v listu a najde nejbližší bod zadanému bodu, který zároveň nebude totožný. Metoda vrací index nejbližšího bodu.

Metoda *getDelaunayPointIdx* má na vstupu hranu Edge a list bodů QPoint3D. Metoda projde všechny vstupní body (kromě bodů identických s body definujícími hranu) a vybere bod z levé poloroviny, který spolu s body hrany vytváří kružnici o nejmenším poloměru. Funkce vrací index optimálního bodu. Pokud není nalezen optimální bod, je vrácena hodnota -1. Bližší popis algoritmu je uvedený v kapitole 3.5.1.

Metoda *DT* má na vstupu list bodů QPoint3D, ze kterých vytváří Delaunayho triangulaci. Tato metoda vrací list jednotlivých hran *Edge*. Podrobný popis algoritmu je uveden v kapitole 3.5.2.

Metoda *updateAEL* má na vstupu hranu Edge a list hran. Tato metoda provádí aktualizaci listu AEL a to tak, že když se vstupní hrana nachází v listu AEL, je z něj odstraněna. Když se vstupní hrana v listu nenachází je prohozena její orientace a tato hrana je přidána do listu AEL.

Metoda *getCLpoint* má na vstupu 2 body QPoint3D a hodnotu nadmořské výšky *z*. Výstupem funkce je bod QPoint3D, který se nachází na hraně definované 2 vstupními body a je v nadmořské výšce *z*.

Metoda *createCL* má na vstupu výsledný list hran *Edge* po triangulaci, minimální a maximální hodnotu z a krok tvorby vrstevnic. Tato metoda v trojúhelníkové síti vytvoří vrstevnice v určeném intervalu výšek a s definovaným krokem. Výstupem je list hran *Edge*. Podrobný popis algoritmu je popsán v kapitole 3.5.3.

Metoda *slope* výsledný list hran *Edge* po triangulaci a velikost měřítka, jakým byla data zmenšena při nahrávání do programu, aby bylo možné vypočítat sklon korektně. Pro každý trojúhelník triangulace se vypočítá velikost sklonu. Metoda vrací list sklonů, přičemž pořadí odpovídá pořadí po triangulaci. Jelikož trojúhelníky jsou definovány třemi hranami, tak 3 hrany jdoucí za sebou ve vstupním listu definují jeden trojúhelník triangulace. Bližší popis algoritmu je uvedený v kapitole 3.5.4.

Metoda *aspect* má na vstupu výsledný list hran *Edge* po triangulaci. Metoda vrací list orientací jednotlivých trojúhelníků. Výstup je uspořádán stejně jako u metody *slope*. Bližší popis algoritmu je popsán v kapitole 3.5.5.

Metoda *meanHeight* má na vstupu výsledný list hran *Edge* po triangulaci. Výstupem je list, ve kterém jsou uloženy průměrné nadmořské výšky jednotlivých trojúhelníků triangulace. Výstup je uspořádán stejně jako u metody *slope*.

3.5. Bližší popis vybraných algoritmů

3.5.1. getDelaunayPointIdx

Inicializuj index idx optimálního bodu na -1

Inicializuj minimální poloměr kružnice na nekonečno

Počáteční bod hrany e inicializuj jako bod p1

Koncový bod hrany e inicializuj jako bod p2

***Projdi** všechny body vstupní množiny*

    ***Když** je bod stejný jako jeden z bodů p1 nebo p2 definující hranu:*

        Pokračuj na další bod

    ***Když** se bod nachází v levé polorovině hrany:*

        Zjisti střed a poloměr kružnice definované aktuálním bodem a body p1, p2

        ***Když** je střed kružnice v pravé polorovině hrany:*

            Poloměru této kružnice změníme znaménko

        ***Když** je poloměr aktuální kružnice menší než minimální poloměr kružnice:*

            Aktualizuj minimální poloměr kružnice

            Aktualizuj index bodu idx

***Vrať** idx*

3.5.2. DT

Inicializuj prázdný list hran po triangulaci dt

Inicializuj prázdný pomocný list ael

Inicializuj první bod q jako bod s minimální souřadnicí x

Najdi nejbližší bod q_n k bodu q

Z bodů q a q_n vytvoř hranu e_1

Najdi optimální Delaunayho bod p vůči hraně e_1

***Když** není nalezen žádný optimální bod:*

Změň orientaci hrany e_1

Najdi optimální Delaunayho bod p vůči hraně e_1

Vytvoř 3 hrany trojúhelníku definovaného body q, q_n a p

Přidej hrany do listů dt a ael

***Dokud** není ael prázdný*

Vezmi poslední hranu e_1 z ael

Změň orientaci e_1

Najdi optimální Delaunayho bod p vůči hraně e_1

***Když** je nalezen bod p:*

Vytvoř hrany trojúhelníku definovaného bodem p a hranou e_1 jako e_1, e_2, e_3

Přidej e_1, e_2, e_3 do dt

Změň orientaci e_2, e_3

***Když** se e_2 a e_3 nachází v ael:*

Odeber e_2, e_3 z ael

Jinak

Změň orientaci e_2, e_3

Přidej e_2, e_3 do ael

3.5.3. createCL

Inicializuj prázdný list cl s vrstevnicemi

***Projdi** všechny trojúhelníky v listu triangulace dt*

Získej vrcholové body p1, p2, p3 jednotlivých trojúhelníků

Získej výšky bodů p1, p2, p3

***Projdi** roviny v intervalu z_min, z_max s krokem dz*

***Když** trojúhelník leží v rovině z:*

Pokračuj na další trojúhelník

***Jinak když** je jedna hrana trojúhelníku v rovině z:*

Přidej do cl

***Jinak když r**ovina protíná dvě hrany:*

Spočítej průsečíky A, B obou hran a roviny z

Vytvoř hranu e spojující body A, B

Přidej hranu e do cl

***Vrať** cl*

3.5.4. slope

Inicializuj prázdný list sklonů slope

***Projdi** všechny trojúhelníky v listu dt*

Získej vrcholy trojúhelníků

Spočítej vektory u, v plochy definované vrcholy trojúhelníku

Spočítej normálový vektor roviny trojúhelníku jako vektorový součin u, v

***Když** je souřadnice c normálového vektoru menší než 0:*

Změň znaménka všech složek normálového vektoru

Spočítej velikost normálového vektoru

Vypočítej sklon ve stupních

Přidej sklon do listu slope

***Vrať** slope*

3.5.5. aspect

Inicializuj list expozice aspect

***Projdi** všechny trojúhelníky v listu dt*

Získej vrcholy trojúhelníků

Spočítej vektory u, v plochy definované vrcholy trojúhelníku

Spočítej normálový vektor roviny trojúhelníku jako vektorový součin u, v

Spočítej azimut průmětu normálového vektoru do roviny xy

Přidej azimut do listu aspect

***Vrať** aspect*

3.6. Vstup

Vstupní data jsou ve formátu *.shp* a do uživatelského rozhraní jsou nahrávána pomocí tlačítka Open. Jedná se o body se souřadnicemi x a y. Souřadnice z je získána z atributové tabulky, kde se musí nacházet ve čtvrtém sloupci a data jsou ve formátu *float*.

3.7. Výstup

Výstup je zajištěn v podobě vizualizace v uživatelském rozhraní. Vizualizace vrstevnic je vytvořena celkem dvěma způsoby – základní vrstevnice jsou vizualizovány červeně a zvýrazněné vrstevnice zase tučně červeně. Výstupy funkce slope jsou vizualizovány barevnou stupnicí podle velikosti sklonu, jejíž intervaly byly zvoleny následovně (viz tabulka 1).

Výstupy funkce *aspect* jsou také znázorněny barevnou stupnicí. Ta je zvolena dle orientace svahu podle světových stran (viz tabulka 2). Pro vizualizaci barevné hypsometrie byly vybrány barvy, jež dle názoru autorek nejlépe korespondují s příslušnými nadmořskými výškami. Tato stupnice je uvedena v tabulce 3.

Tabulka 1: Intervaly a kódy barevné stupnice pro funkci slope

Barva RGB	Sklon [v °]
Bílá (255, 255, 255)	Méně než 0.50
Světle šedá (230, 230, 230)	0.50–1.50
Světle šedá (200, 200, 200)	1.51–3.00
Světle šedá (180, 180, 180)	3.01–5.00
Šedá (130, 130, 130)	5.01–15.00
Tmavě šedá (90, 90, 90)	15.01–30.00
Tmavě šedá (50, 50, 50)	30.01–45.00
Černá (10, 10, 10)	Více než 45.00

Tabulka 2: Intervaly a kódy barevné stupnice pro funkci expozice

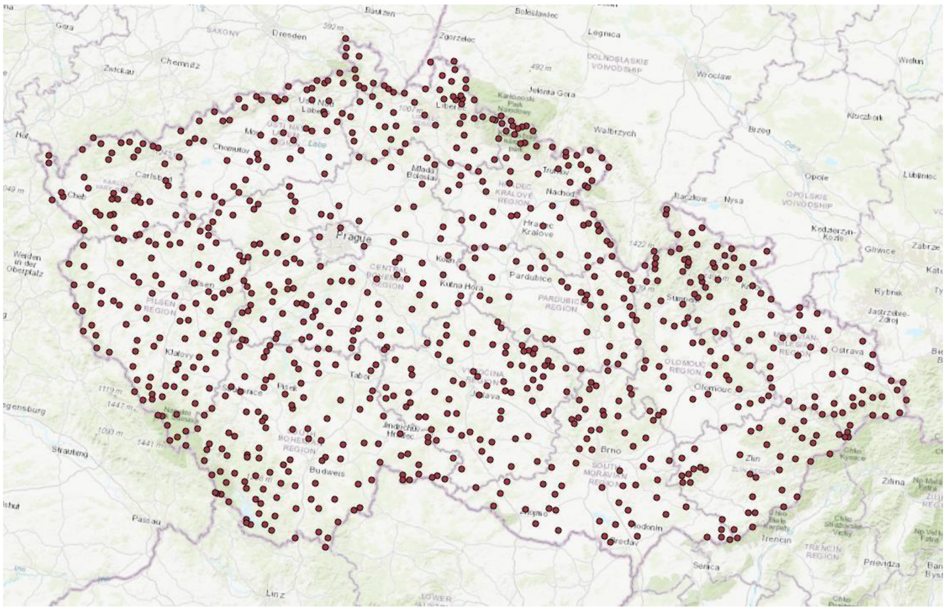
Barva RGB	Světová strana
Červená (255, 0, 0)	S
Oranžová 255, 165, 0)	SV
Žlutá (255, 255, 0)	V
Zelená (0, 255, 0)	JV
Světle modrá (64, 224, 208)	J
Modrá (0, 0, 255)	JZ
Tmavě modrá (0, 0, 128)	Z
Fialová (255, 0, 255)	SZ

Tabulka 3: Intervaly a kódy barevné stupnice pro barevnou hypsometrii

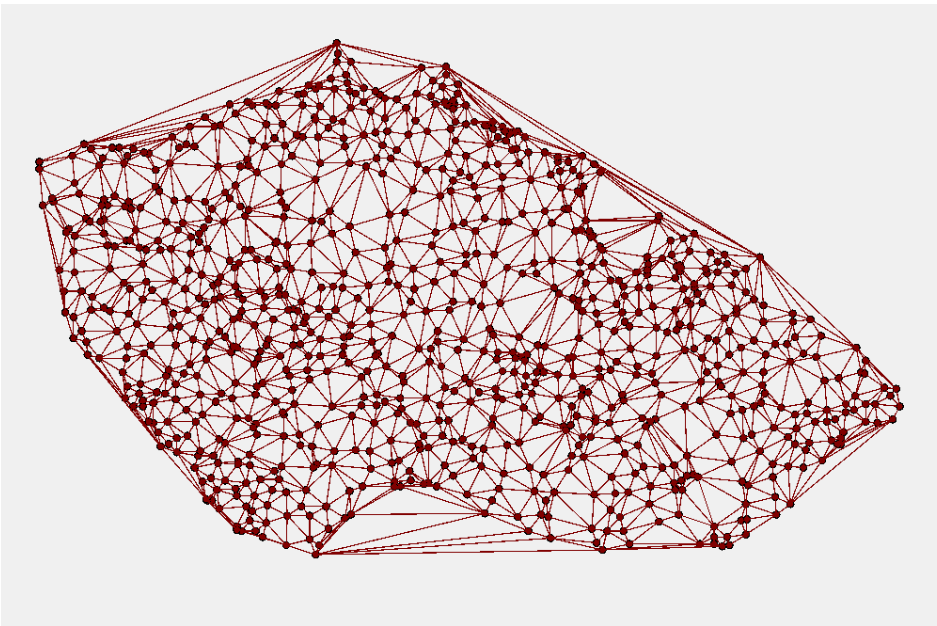
Barva RGB	Nadmořská výška
Světle modrá (0, 255, 274)	Méně než 200.1
Tyrkysová (0, 255, 179)	200.1–250.0
Světle zelená (0, 255, 111)	250.1–300.0
Světle zelená (0, 255, 85)	300.1–350.0
Světle zelená (26, 255, 0)	350.1–400.0
Světle zelená (77, 255, 0)	400.1–450.0
Světle zelená (102, 255, 0)	450.1–500.0
Světle zelená (171, 255, 0)	500.1–550.0
Žlutozelená (222, 255, 0)	550.1–600.0
Žlutá (255, 247, 0)	600.1–650.0
Tmavě žlutá (255, 213, 0)	650.1–700.0
Světle oranžová (255, 171, 0)	700.1–750.0
Oranžová (255, 128, 0)	750.1–800.0
Tmavě oranžová (255, 94, 0)	800.1–850.0
Červená (255, 0, 0)	850.1–900.0
Tmavě hnědá (101, 0, 0)	Více než 900.0

4. Výsledky

Pro otestování základní funkcionality aplikace byla zvolena množina bodů, které představují vrcholy nacházející se na území ČR (obrázek 1). Prvním krokem bylo vyzkoušet funkčnost a věrohodnost funkce Delaunay triangulace. Funkčnost této části byla klíčová, protože se od ní odvíjelo fungování následujících analýz. Výsledek je na obrázku 2.

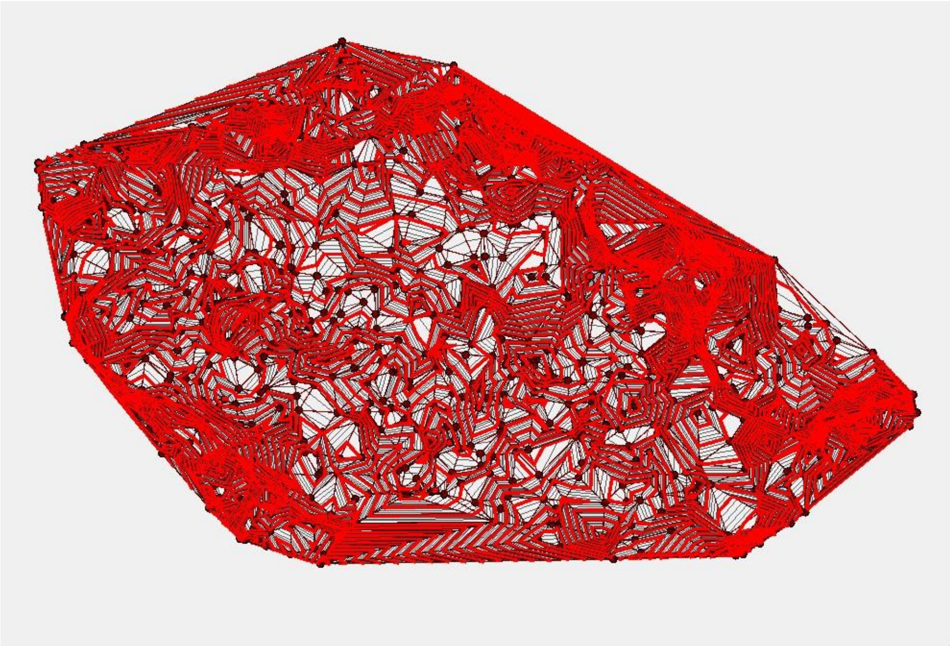


Obrázek 1: Testovací vstupní množina bodů vrcholů ČR



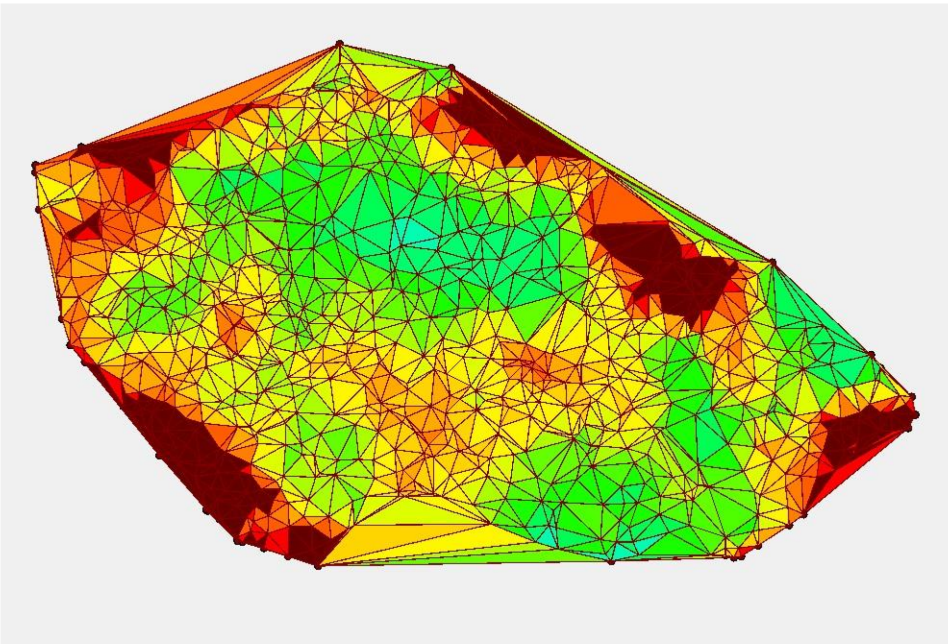
Obrázek 2: Vytvoření Delaunay triangulace nad testovací množinou

Pro vytvoření adekvátní reprezentace vrstevnic bylo nutné změnit prvotní nastavení. S defaultním nastavením vrstevnic funkce nepokryla celé území ČR, čímž vznikly pouhé shluky bodů, mezi kterými byla funkce tvorby vrstevnic aktivována. Pro výsledek, jenž je k vidění na obrázku 3, bylo potřeba nastavit krok vrstevnic 10 m s minimální nadmořskou výškou 0 m a maximální zase 2 000 m, pro pokrytí celého území.



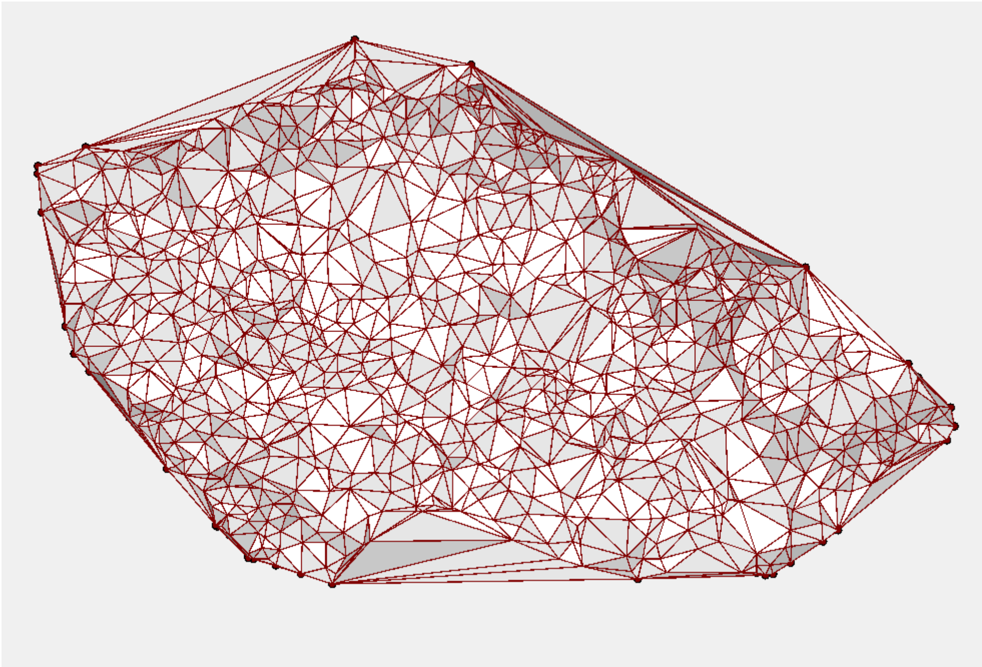
Obrázek 3: Vytvoření vrstevnic nad testovací množinou

Další analýzou, kterou bylo možno nad vstupní vrstvou vytvořit byla barevná hypsometrie. Ta tvoří 2D DMT pomocí barevné stupnice, jejíž barvy byly přiděleny jednotlivým trojúhelníkům vytvořených DT, dle odpovídající nadmořské výšky. Jednotlivé intervaly této barevné stupnice jsou uvedeny v tabulce 3. Věrohodnost této metody je poměrně vysoká a věrně odpovídá skutečnosti.

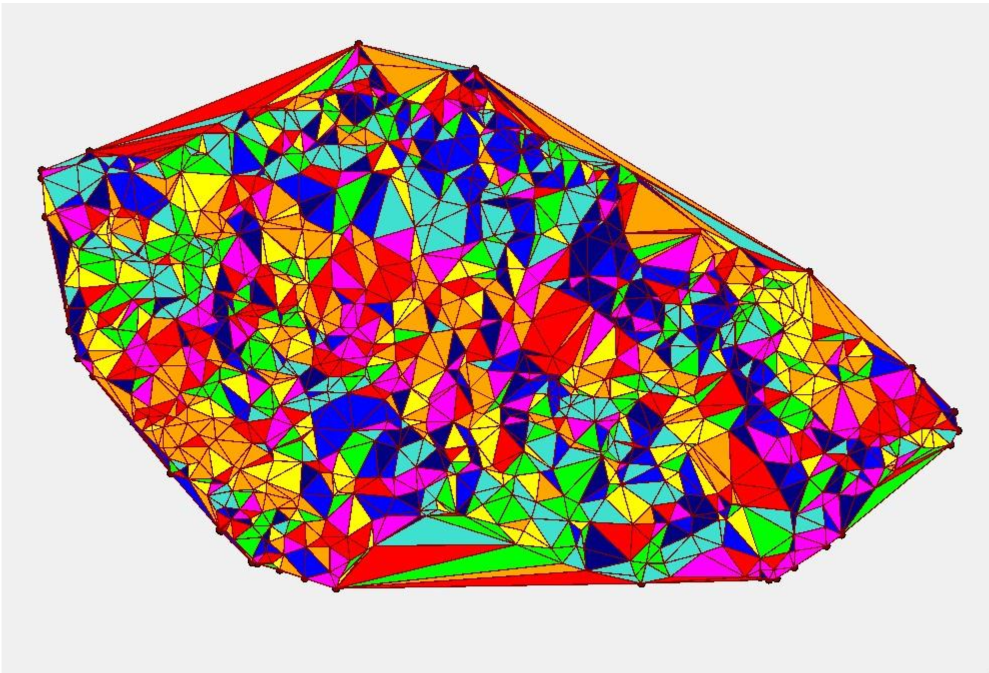


Obrázek 4: Vytvoření barevné hypsometrie nad testovací množinou

Je možno vytvořit i následující DTM analýzu sklon (obrázek 5). Ten analyzuje sklon veškerých trojúhelníků, který je vizualizován monochromatickou barevnou stupnicí (viz tabulka 1). Poslední vytvořená analýza je analýza orientace svahu vůči světovým stranám (viz obrázek 6). Tato analýza využívá vytvořenou barevnou stupnici, která je uvedena v tabulce 2.

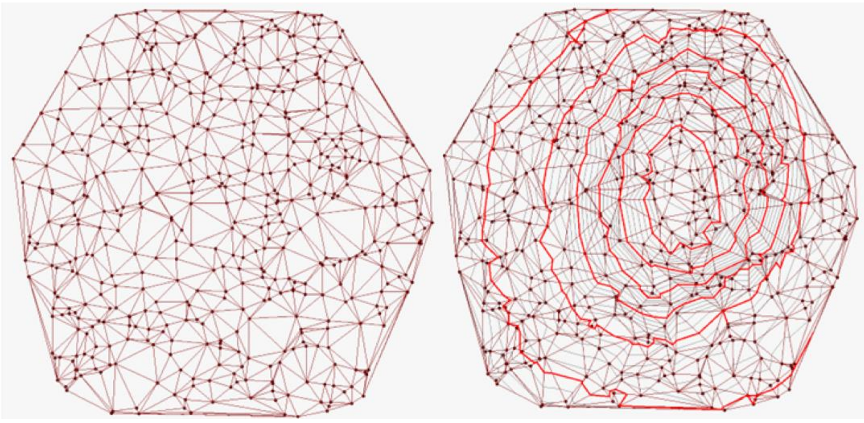


Obrázek 5: Vytvoření sklonu nad testovací množinou

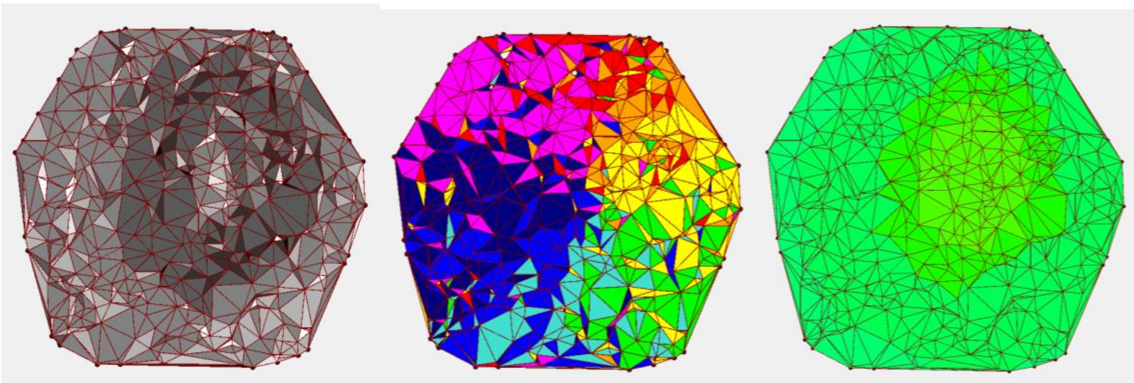


Obrázek 6: Vytvoření orientace nad testovací množinou

Dále byla aplikace testována na datech reprezentující přírodní tvary (kupa, údolí, hřbet). Při triangulaci u všech tvarů nevznikl žádný viditelný problém. V případě kupy bylo generování vrstevnic i vizualizace sklonu a orientace v pořádku (viz obrázky 7 a 8).

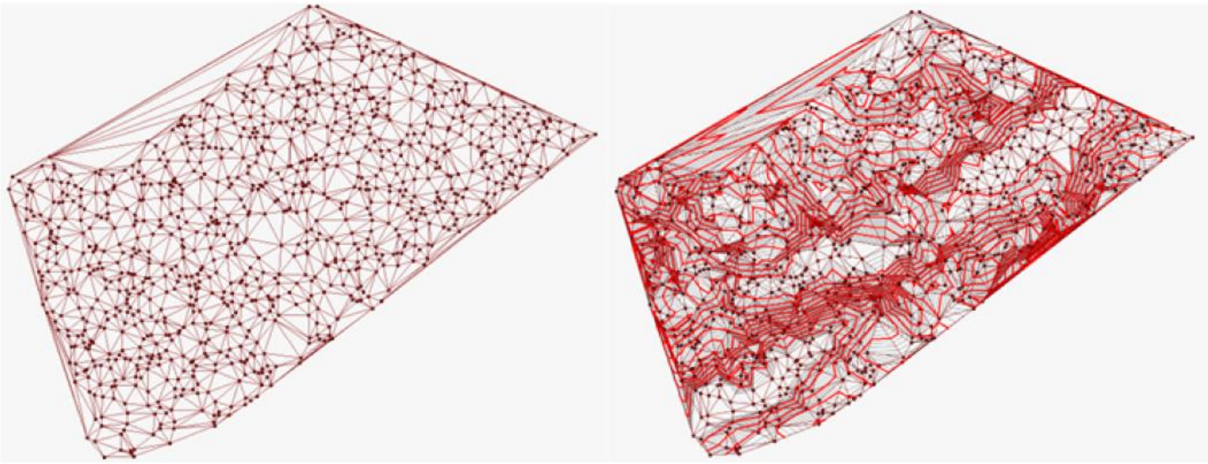


Obrázek 7: Triangulace (vlevo) a generování vrstevnic (vpravo) pro kupu

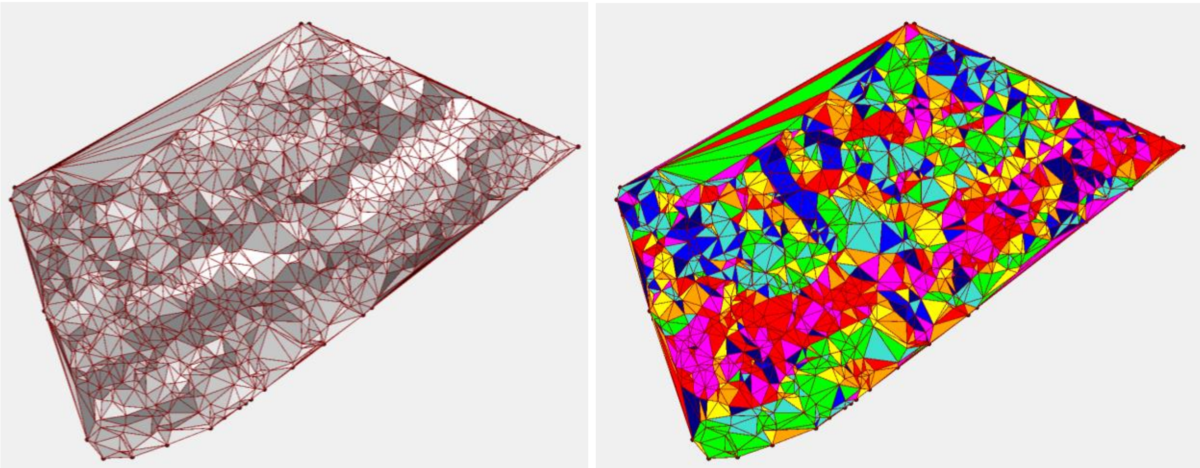


Obrázek 8: Sklon (vlevo), orientace svahů (uprostřed) a barevná hypsometrie (vlevo) pro kupu

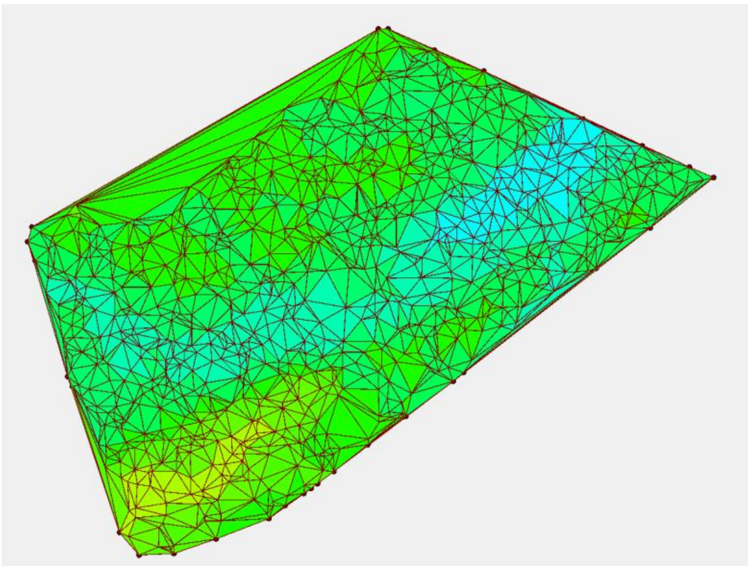
V případě údolí funguje triangulace, tvorba vrstevnic i vizualizace sklonu bez větších potíží (obrázek 9 a 10). Problematický je výpočet orientace svahu pro dno údolí. Zde je svah blízký 0 a proto je orientace těžce spočitatelná a skokově se mění u každého trojúhelníku (obrázek 10). Barevná hypsometrie je vidět na obrázku 11.



Obrázek 9: Triangulace (vlevo) a generování vrstevnic (vpravo) pro údolí

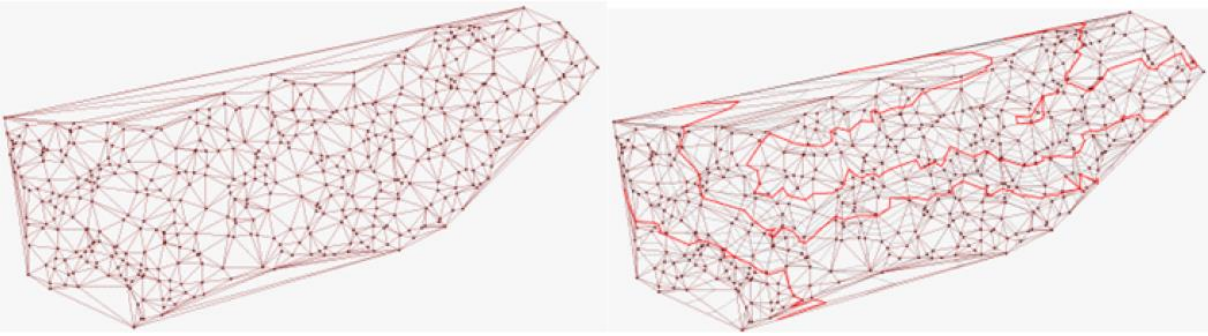


Obrázek 10: Sklon (vlevo) a orientace svahů (vpravo) pro údolí

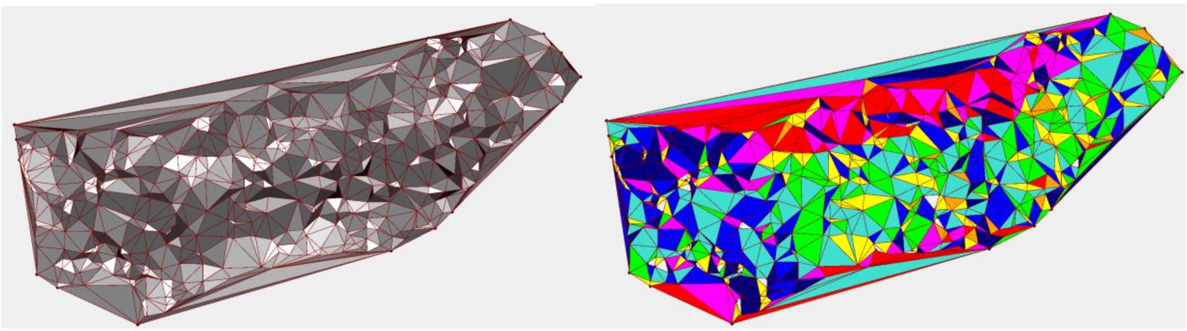


Obrázek 11: Barevná hypsometrie pro údolí

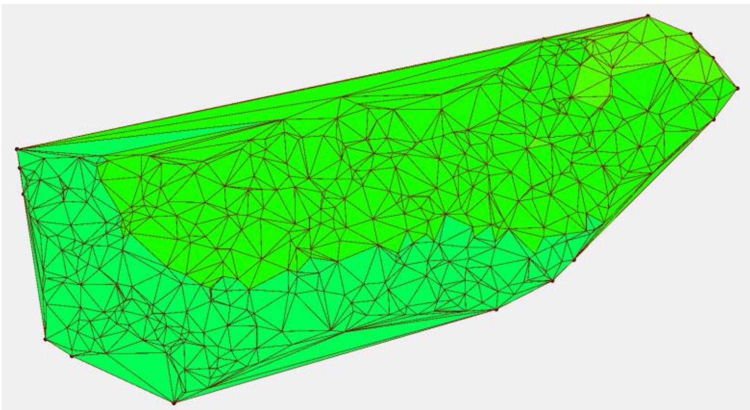
V případě hřbetu dopadly výsledky podle vizuálního zhodnocení nejhůře. Objevuje se zde stejný problém jako u údolí – orientace je na vrcholu hřbetu těžce spočitatelná, a proto je výsledkem „barevná mozaika“. Výsledky jsou uvedeny na obrázku 12, 13 a 14.



Obrázek 12: Triangulace (vlevo) a generování vrstevnic (vpravo) pro hřbet



Obrázek 13: Sklon (vlevo) a orientace svahů (vpravo) pro hřbet



Obrázek 14: Barevná hypsometrie pro hřbet

5. Závěr

Vytvořená aplikace vytváří digitální model reliéfu nad vstupními daty pomocí Delaunay triangulace. Vstupní množina dat je libovolně nahrávatelná vrstva bodového mračna, jež je programu předávána v podobě shapefile. Aplikace rovněž umí vytvářet analýzy terénu, jako jsou sklon, orientace terénu, vytvořit terénní vrstevnice anebo barevnou hypsometrii.

Mezi jednotlivými nástroji aplikace funguje předepsaná hierarchie – například pokud by uživatel chtěl vykreslit vrstevnice, je nutné prvně vytvořit Delaunay triangulaci, aby tvorba vrstevnic byla, jakkoliv možná. Tímto mezikrokem se však snižuje uživatelská přívětivost aplikace, proto možným vylepšením by mohlo být zahrnutí DT do všech kroků a analýz, aby se tím tak snížil počet mezikroků. Možným rozšířením by také mohl být názorný popis nadmořských výšek jednotlivých bodů. Dále by bylo možné vylepšit algoritmus triangulace tak, aby vytvářel triangulaci nekonvexní oblasti vymezené polygonem (např. polygon území Česka), čímž by zmizely některé krajní trojúhelníky a trojúhelníková síť by lépe zachovávala tvar území.

6. Použité zdroje

BAYER, T. (2008): Algoritmy v digitální kartografii. Univerzita Karlova v Praze, Nakladatelství Karolinum, Praha, 251 s.

BAYER, T. (2023): Rovinné triangulace a jejich využití. Prezentace k předmětu *Algoritmy počítačové kartografie*. Dostupné z: http://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk5_new.pdf (cit. 13.5.2023).