Univerzita Karlova
Přírodovědecká fakulta

Algoritmy počítačové kartografie

Technická zpráva k třetí semestrální úloze

Digitální model terénu

David Šklíba a Filip Zadražil
1. N-GKDPZ
Praha 2022

1 Zadání a údaje o bonusových úlohách

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

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

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

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

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

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

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

Zhodnocení činnosti algoritmu včetně ukázek proveďte alespoň na **3 strany** formátu A4.

Hodnocení:

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

V rámci vzniklého programu nebyly řešeny **žádné** bonusové úlohy.

2 Popis a rozbor problému

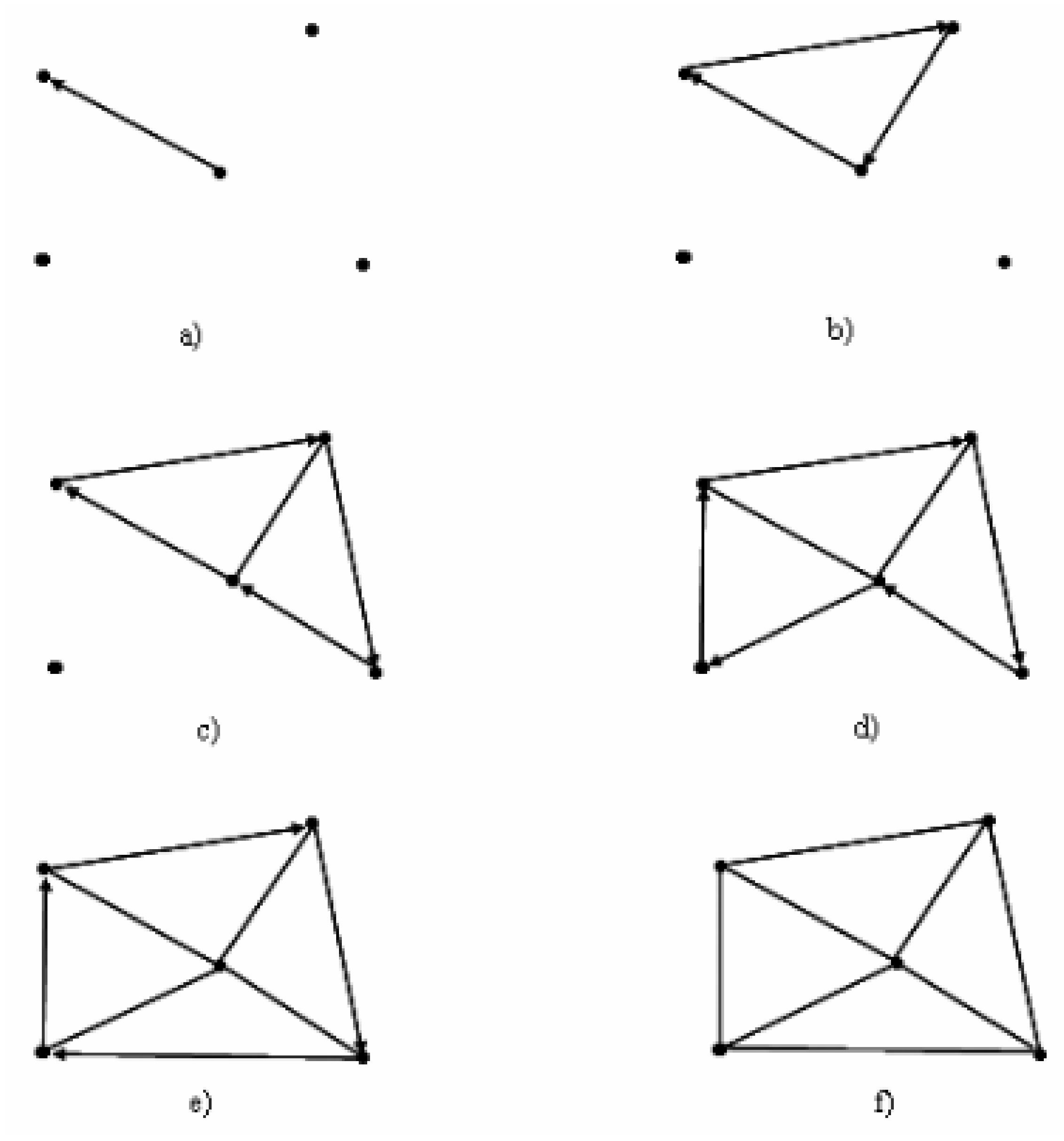
2.1 Delaunayho triangulace

Delaunayho triangulace (DT) je jednou z nejčastěji používaných metod triangulace v GIS. Pracuje na maximalizaci minimálního vnitřního úhlu vytvářených trojúhelníků. Jedná se o kritérium globální, tedy podle uvedeného kritéria se optimalizují geometrické parametry všech trojúhelníku v síti. Princip této metody tkví v tom, že v kružnici opsané libovolnému trojúhelníku z DT neleží žádný další bod ze vstupní množiny. Existuje navíc hned několik upravených verzí algoritmu. Například v tomto programu byla konkrétně využita metoda inkrementální konstrukce DT.

Tato metoda, využívající inkrementální vkládání, je založena na postupném přidávání bodů do již vytvořeného DT. Nejprve je nutné vybrat libovolný bod z množiny a najít jeho nejbližší bod (v euklidovském prostoru). Tento krok dá vzniknout orientované hraně $e = (P_1, P_2)$. Následně je hledán bod $\underline{P}$, který musí ležet v levé polorovině od hrany e a zároveň musí minimalizovat poloměr r kružnice k, která je opsaná tomuto bodu a této hraně. Vzorcem lze tuto podmínku zapsat následovně:

$$\underline{P} = arg\ min\ r'(k_i)_{\forall P_i \in \sigma_L(e)}, \quad k_i = (e, P_i), \quad P_i(e) \in \sigma_L$$

Po nalezení na základě podmínek nejvhodnějšího bodu vzniknou nové hrany $e_1 = (P_2, \underline{P}), e_2 = (\underline{P}, P_1)$. Pokud bod P nebyl nalezen otočíme orientaci hrany a vyhledání bodu se opakuje, opět v levé polorovině hrany e. Tím je dokončena základní inicializace algoritmu, která se poté opakuje.



Obrázek 1: Princip metody inkrementální konstrukce DT

Po nalezení vhodného Dealuneyho bodu jsou dvě nové hrany společně s první definovanou hranou e vloženy do seznamu aktivních hran (AEL). Následně se vezme první hrana, otočí se její orientace a nalezne se pro ni bod $\underline{P}$. Dvě nově vzniklé hrany poté přidáme do AEL za předpokladu, že se nově vytvořené hrany v AEL již nenachází, byť s opačnou orientací. V tomto případě je hrana z AEL odstraněna a přidána do výsledné triangulace. Pokud pro aktuální hranu nebyl nalezen žádný bod znamená to, že hrana je součástí konvexního obalu a takovou také přidáme do výsledné triangulace. Pokud nedojde k nalezení bodu $\underline{P}$, bod je vyhledáván v pravé polorovině.

Základní princip metody inkrementální konstrukce je naznačen ve schématu na Obrázku 1.

2.2 Konstrukce vrstevnic lineární interpolací

Problém této úlohy spočívá v nalezení průsečnice roviny určené trojúhelníkem, který je součástí DT a vodorovné roviny ρ o výšce h. Tento proces se opakuje u všech trojúhelníků, které jsou součástí DT. Výpočet souřadnic bodů průsečnice AB je prováděn pouze pro variantu, kdy průsečnice je úsečkou. Vzorec pro výpočet souřadnic má tento tvar:

$$x_a = \frac{x_3 - x_1}{z_3 - z_1}(z - z_1) + x_1 \qquad x_b = \frac{x_2 - x_1}{z_2 - z_1}(z - z_1) + x_1,$$
$$y_a = \frac{y_3 - y_1}{z_3 - z_1}(z - z_1) + y_1 \qquad y_b = \frac{y_2 - y_1}{z_2 - z_1}(z - z_1) + y_1,$$

Kontrolu, zda rovina ρ prochází hranou (p_i, p_{i+1}), lze pomocí rovnice

$$(z - z_i)(z - z_{i+1}) < 0.$$

Níže v *Algorithm 1* je pomocí pseudokódu konstrukce vrstevnic podrobně popsána.

Algorithm 1 Konstrukce vrstevnic lineární interpolací

1: Vytvoř prázdný seznam hran cl
2: Pro každou po sobě jdoucí trojici hran $e_1 = (p_1 p_2)$, $e_2 = (p_2 p_3)$, $e_3 = (p_3 p_1)$ Delaunayho triangulace:
3:     $p_i = [x_i, y_i, z_i]$
4:     Pro každou hodnotu z z intervalu (z_{min}, z_{max}) s krokem dz:
5:         $dz_i = z_i - z$
6:         $t_{ij} = dz_i dz_j$
7:         **if** $dz_i = dz_{i+1} = dz_{i+2} = 0$:
8:             Jdi na 4)
9:         **else if** $dz_i = dz_{i+1} = 0$:
10:             Přidej hranu e_1 do cl
11:         **else if** $dz_{i+1} = dz_{i+2} = 0$:
12:             Přidej hranu e_2 do cl
13:         **else if** $dz_{i+2} = dz_i = 0$:
14:             Přidej hranu e_3 do cl
15:         **else if** $t_{12} \leq$ and $t_{23} < 0$ **or** $t_{12} < 0$ and $t_{23} \leq 0$:
16:             A=bod ležící na hraně e_1 ve výšce z
17:             B=bod ležící na hraně e_2 ve výšce z
18:             $e = (A, B)$
19:             přidání hrany e do cl
20:         **else if** $t_{23} \leq$ and $t_{31} < 0$ **or** $t_{23} < 0$ and $t_{31} \leq 0$:
21:             B=bod ležící na hraně e_2 ve výšce z
22:             C=bod ležící na hraně e_3 ve výšce z
23:             $e = (B, A)$
24:             přidání hrany e do cl
25:         **else if** $t_{12} \leq$ and $t_{31} < 0$ **or** $t_{12} < 0$ and $t_{31} \leq 0$:
26:             C=bod ležící na hraně e_3 ve výšce z
27:             A=bod ležící na hraně e_1 ve výšce z
28:             $e = (C, A)$
29:             přidání hrany e do cl

2.3 Analýza sklonu terénu

Nad digitálním modelem terénu lze následně provádět různé analytické úlohy. Jednou z nich je analýza sklonu terénu. Využití nachází například při analýze hydrologických poměrů, půdních sesuvů, detekci lavin, nebo při návrzích výstavby komunikací či jiných stavebních objektů. Výpočet sklonu se provádí nad každým trojúhelníkem DMT.

Rovnice roviny ρ, pro níž je sklon počítán, je

$$\rho = ax + by + cz + d = 0$$

,

pro níž je počítán gradient $(\nabla\rho)$ pomocí vzorce

$$\nabla\rho(x_0, y_0, z_0) = (\frac{\sigma_\rho}{\sigma_x}(x_0), \frac{\sigma_\rho}{\sigma_y}(y_0), (\frac{\sigma_\rho}{\sigma_z}(z_0)) = (a, b, c).$$

Rovinu ρ lze definovat touto maticí

$$\begin{vmatrix} x - x_1 & y - x_1 & z - x_1 \\ x_2 - x_1 & y_2 - x_1 & z_2 - x_1 \\ x_3 - x_1 & y_3 - x_1 & z_3 - x_1 \end{vmatrix} = 0,$$

z níž se následně výpočítá odchylka φ od roviny π vzorcem

$$\varphi = arccos\left|\frac{n_1 n_2}{||n_1||||n_2||}\right|$$

.

Algoritmus aplikovaný v programu pro výpočet sklonu je popsán níže pomocí pseudokódu.

Algorithm 2 Výpočet sklonu

$e_1 = (p_1p_2)$, $e_2 = (p_2p_3)$, $e_3 = (p_3p_1)$ - hrany trojúhelníku DT
$p_i = [x_i, y_i, z_i]$
$\vec{u} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$
$\vec{v} = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$
$\vec{n} = \vec{u} \times \vec{v}$
$\varphi = \arccos\left(\frac{n_z}{|\vec{n}|}\right)$

2.4 Analýza orientace terénu

Druhou analyzovanou vlastností je orientace terénu. Ta je definována jako azimut průmětu gradientu $\nabla\rho$ do roviny x, y. Vektor gradientu lze označit např. písmenem v, pro nějž platí

$$v = (\frac{\sigma_\rho}{\sigma_x}(x_0), \frac{\sigma_\rho}{\sigma_y}(y_0), 0) = (a, b, 0).$$

Pro azimut vektoru v pak platí vzorec

$$A = arctan(\frac{b}{a})$$

.

Výpočet orientace se opět provádí pro každý jeden trojúhelník v DMT.

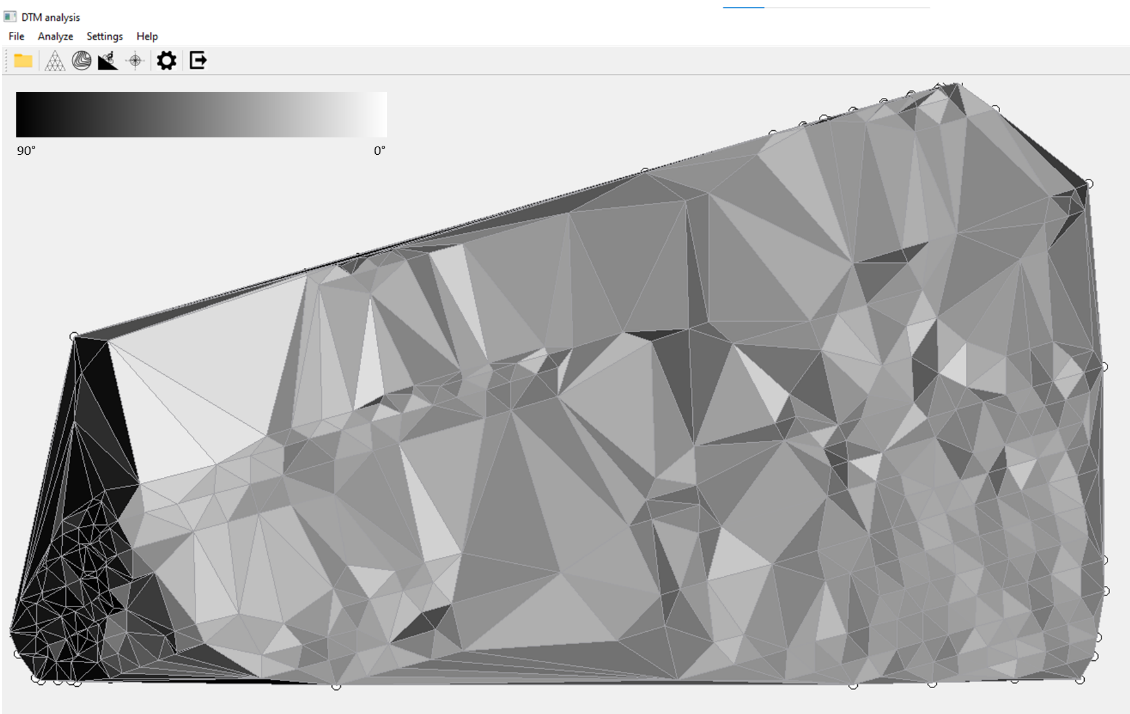
Algorithm 3 Výpočet orientace svahu

$e_1 = (p_1p_2)$, $e_2 = (p_2p_3)$, $e_3 = (p_3p_1)$ - hrany trojúhelníku DT
$p_i = [x_i, y_i, z_i]$
$\vec{u} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$
$\vec{v} = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$
$\vec{n} = \vec{u} \times \vec{v}$
if $n_x \neq 0$:
    $A = \arctan\left(\frac{n_y}{n_x}\right)$
else:
    $A = \emptyset$
if $n_x > 0$:
    $A = 90 - A$
else if $n_x < 0$:
    $A = 270 - A$

3 Vstupní a výstupní data

Data vstupující do programu tvoří množina bodů, která představuje model terénu. Výstupem je polyedrický DMT nad vstupní množinou bodů představovaný vrstevnicemi a je doplněn o vizualizaci sklonu trojúhelníků a jejich expozici.

4 Ukázka vytvořené aplikace



Obrázek 2: Ukázka vytvořené aplikace - widget

Na Obrázku 2 je widget, v němž můžeme vytvořený program spustit. V horní liště se uživateli nabízí několik možností postupu práce. Levá ikona složky nám umožňuje nahrání dat do programu ve formátu *.txt*. Vedle ní se postupně nacházejí ikony na tvorbu Delauneyho triangulace, tvorbu vrstevnic, analýzu sklonu a analýzu expozice. Ikona ozubeného kolečka označuje možnost Options, v níž nalezneme pole pro tři vybrané veličiny, jejichž hodnoty si může uživatel sám nastavit dle vlastních preferencí. Obsahují nastavení minimální a maximální vykreslované hodnoty vrstevnic a interval, po němž se příslušná vrstevnice vykreslí. Ikona zcela vpravo ukončí program. V případě, že uživatel nemá k dispozici vhodná data pro používané analýzy, lze je vlastnoručně naklikat pomocí kurzoru myši. V hlavním okně widgetu je pak vidět model DMT včetně analýzy sklonu svahů.

Ukázka je dodatečně doplněna o legendu pro analýzu sklonu, pro níž byla použita spojitá stupnice šedi, v níž bílá barva odpovídá rovině (0°) a černá barva stěně (90°).

5 Dokumentace

Skript *mainform.py*, přes který se program spouští, obsahuje třídu uživatelského rozhraní *Ui_mainform* navrženého pomocí SW *QTCreator*. Pro jednotlivá tlačítka (ikony) má třída definované metody odkazující na skripty *draw.py* a *algorithms.py*. Skript *draw.py* obsahuje třídu *Draw*, která byla navržena pro vizualizaci objektů ve widgetu uživatelského rozhraní.

Parametry třídy *Draw* jsou:

- seznam *self.points* objektů typu *QPoint3D* vstupních bodů,
- seznam *self.dt* objektů typu *Edge* hran Delaynayho triangulace,
- seznam *self.cont_lines* objektů typu *Edge* segmentů vrstevnic,

- seznam $self.triangles\_slope$ objektů typu $QPolygonF$ trojúhelníku Delaunayho triangulace pro vizualizaci sklonu,
- seznam $self.shades$ hodnot typu $integer$ určujících stupeň šedi,
- seznam $self.triangles\_aspect$ objektů typu $QPolygonF$ trojúhelníku Delaunayho triangulace pro vizualizaci orientace svahu,
- seznam $self.colors$ objektů typu $QColor$ vyskytujících se v legendě pro vizualizaci orientace svahu,
- seznam $self.extent$ obsahující souřadnice minmax boxu celého území,
- seznam $self.canvas\_extent$ obsahující šířku a výšku widgetu v pixelech.

Metodami třídy *Draw* jsou:

- $insertFile$ - načítá vstupní soubor ve formátu *.txt*, ukládá souřadnice jednotlivých bodů do seznamu $self.points$ a hledá souřadnicové extrémy, které ukládá do seznamu $self.extent$,
- $rescaleData$ - škáluje souřadnice polygonů tak, aby se celé území vešlo do okna widgetu,
- $mousePressEvent$ - po kliknutí do okna widgetu na tomto místě přidá bod do seznamu $self.points$ a náhodně vygeneruje jeho výškovou souřadnici v intervalu od 0 do 500,
- $paintEvent$ - vykresluje body, linie a polygony do okna widgetu,
- $getPoints$ - vrací seznam typu $QPoint$ vstupních bodů,
- $getDT$ - vrací seznam typu $Edge$ hran Delaynayho triangulace,
- $setDT$ - parametru $self.dt$ přiřadí seznam hran DT,
- $setCL$ - parametru $self.cont\_lines$ přiřadí seznam vrstevnic,
- $setTrianglesSlope$ - parametru $self.triangles\_slope$ přiřadí seznam polygonů (trojúhelníkovou síť),
- $setTrianglesAspect$ - parametru $self.triangles\_aspect$ přiřadí seznam polygonů (trojúhelníkovou síť),
- $setShades$ - parametru $self.shades$ přiřadí seznam hodnot odstínů šedi,
- $setColors$ - parametru $self.colors$ přiřadí seznam barev legendy sklonu svahu.

Skript *algorithms.py* obsahuje třídu *Algorithms*, která byla vytvořena pro implementaci vybraných algoritmů.

Metodami třídy *Algorithms* jsou:

- $getPointAndLinePosition$ – určuje pozici bodu vůči přímce,
- $get2LinesAngle$ – počítá velikost úhlu svíraného mezi dvěma úsečkami,
- $getCircleCenterAndRadius$ - počítá střed kružnice a její poloměr definované třemi body,
- $getNearestPointIdx$ - zjišťuje hodnotu indexu nejbližšího bodu k danému bodu,
- $getDelaunayPointIdx$ - hledá k hraně optimální Delaunayho bod,
- DT - pomocí inkrementální metody vytváří síť Delaunayho triangulace,
- $updateAEL$ - aktualizuje seznam aktivních hran,
- $getCLpoint$ – vrací bod o určité hodnotě výšky ležící mezi dvěma body,
- $createCL$ – vytvoří seznam vrstevnic ležících v určitém intervalu s konrétní ekvidistancí,
- *computeSlope* – pro každý trojúhelník Delaunayho triangulace vypočítá hodnotu sklonu,
- *computeAspect* – pro každý trojúhelník Delaunayho triangulace vypočítá hodnotu orientace svahu.

Skript *qpoint3D.py* obsahuje třídu $QPoint3D$ dědící z třídy $QPointF$. Oproti svému rodičovi je obohacena o parametr $self.z$, který vyjadřuje výškovou souřadnici bodu. S ní se váže i jediná metoda, a to $getZ$, která parametru $self.Z$ přiřazuje hodnotu třetí souřadnice.

Skript *edge.py* obsahuje třídu Edge reprezentující orientovanou hranu trojúhelníku. Třída přijímá dva vstupní argumenty, první se ukládá do parametru $self.start$ vyjadřujícího startovní bod hrany, druhý se potom ukládá do paramatru $self.end$ označujícího koncový bod hrany. Metodami třídy $Edge$ jsou:

- $getStart$ - vrací startovní bod hrany $self.start$,

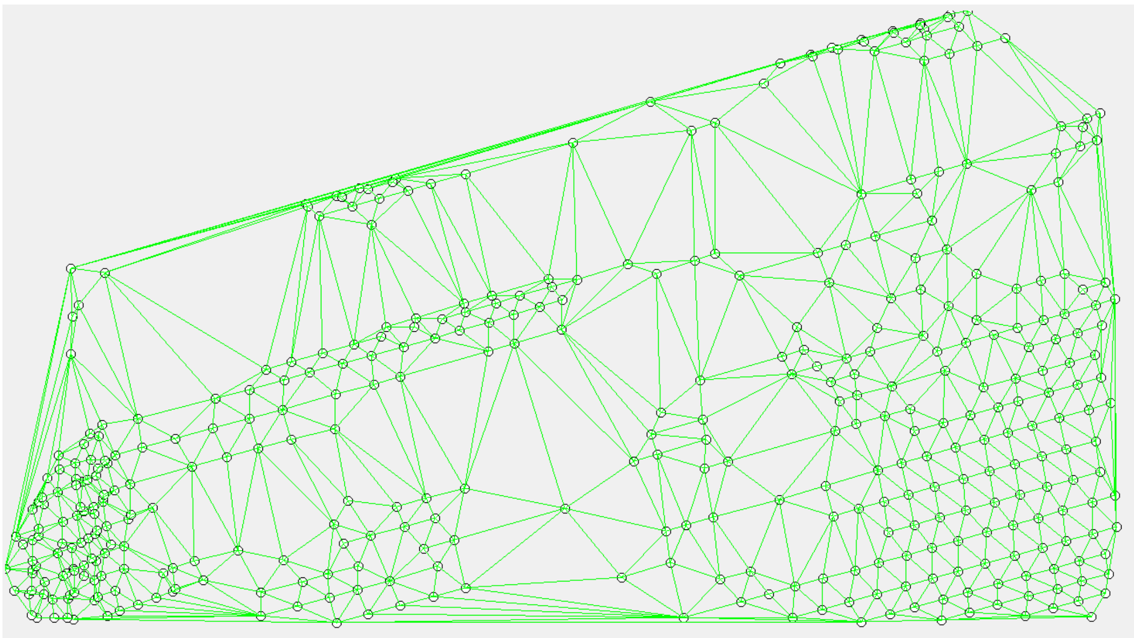
- *getEnd* - vrací koncový bod hrany $self.end$,
- *swithc* - vyměňuje startovní bod $self.start$ s koncovým $self.end$,
- *__eq__* - porovnává shodnost dvou objektů třídy $Edge$.

Skript *triangle.py* obsahuje třídu $Triangle$ reprezentující trojúhelník Delaunayho triangulace. Třída přijímá 3 vstupní parametry, a to jednotlivé objekty hran typu $Edge$, které postupně ukládá do parametrů $self.p1$, $self.p2$, $self.p3$. Metodou třídy $Triangle$ je $getNormalVector$, která počítá normálový vektor trojúhelníku a vrací ho společně s polygonem třídy $QPointF$.

Skript *settings.py* obsahuje třídu uživatelského rozhraní Ui_Dialog, která aplikaci zprostředkovává vyskakující okno, v němž lze nastavit interval a ekvidistanci zobrazení vrstevnic.

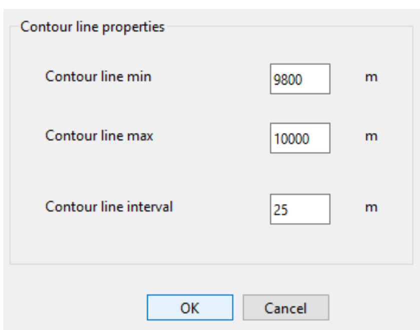
6 Výsledky

Pro získání výsledků byla použita množina bodů z přiloženého textového souboru $input_data.txt$, které byly vytvořeny ručně a následně byly jejich souřadnice uloženy do textového souboru. Vhodná, lepší data pro tvorbu DT se nepodařilo dohledat.

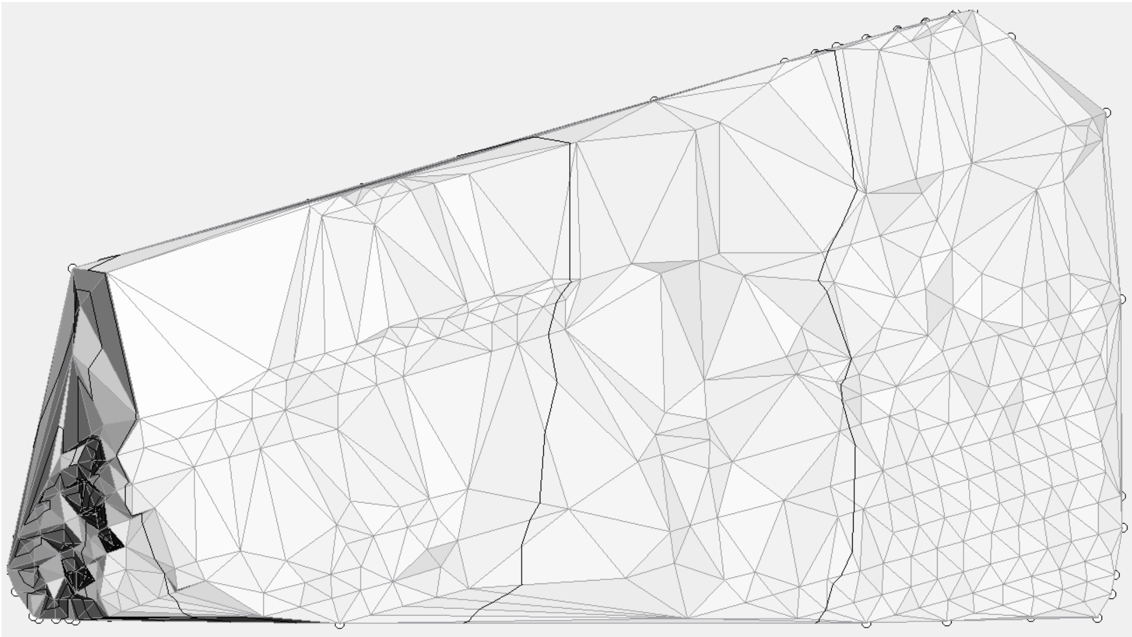


Obrázek 3: Delaunayho triangulace nad vloženou množinou bodů

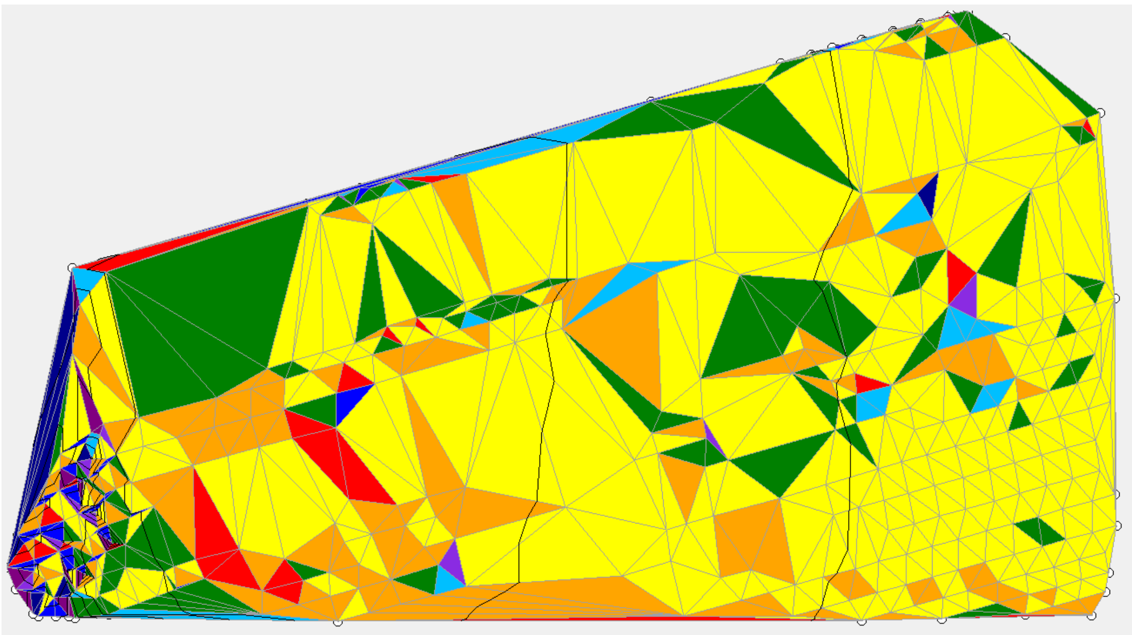
V úvodním kroku je nezbytné vytvořit DT (viz Obrázek 3), neboť z něj se poté odvíjí další úkony. Ačkoli je téměř nemožné z této sítě trojúhelníků cokoli usuzovat, způsob, jakým se vykreslila, je dle našeho názoru velmi uspokojivý. V dalším kroku byl v příslušných trojúhelnících analyzován sklon svahů a zároveň vykresleny vrstevnice (Obrázek 5), pro jejichž vykreslení byly zadány tyto vybrané parametry (viz Obrázek 4).



Obrázek 4: Nastavení parametrů pro vykreslení vrstevnic



Obrázek 5: Analýza sklonu svahů a vykreslení vrstevnic

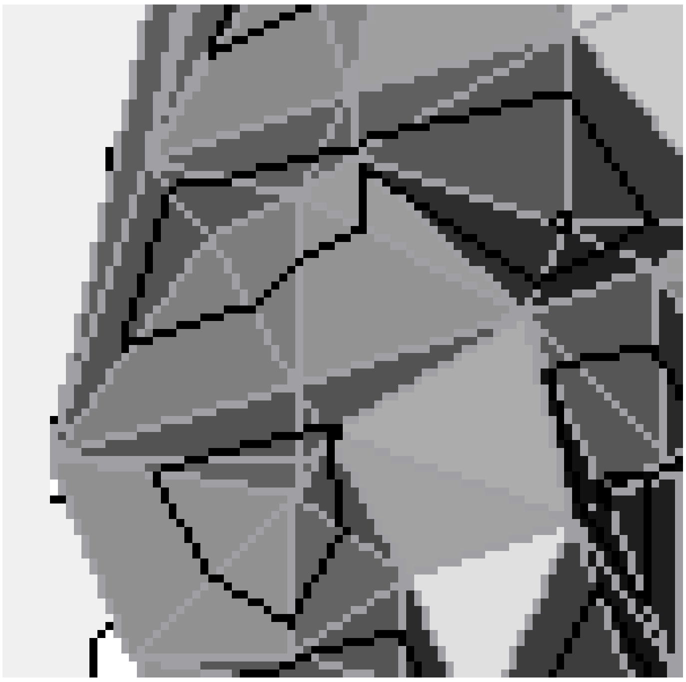


Obrázek 6: Analýza orientace svahů

Z Obrázků 5 a 6 je jasně zřejmé, že vytvořená množina bodů má velmi proměnlivý charakter. Zatímco ve východní části modelu je stoupání velmi pozvolné, v západní části modelu je terén velmi členitý a příkrý. Navíc zatímco v pozvolném svahu je u většiny trojúhelníků navíc zachována i stejná orientace svahu, v pravé části je již svah orientován na různé světové strany a nalezneme zde i několik vrcholů.

I z tohoto ne zcela reprezentativního modelu je však zjevné, že s terénními prvky, jakými jsou například kupa, údolí nebo hřbet nebude mít program problém (viz kopa na Obrázku 7).

Případ analýzy orientace svahů je pak celkem složitý. Je jasné, že orientace v takovémto modelu není příliš snadná a legendu nelze vytvořit na základě jakýchkoli "intuitivních barev"a proto vyžaduje dobrou znalost její legendy. Nakonec bylo vybráno 9 rovnoměrných intervalů dle azimutů a těm byly náhodně přiřazeny barvy (viz Obrázek 8).

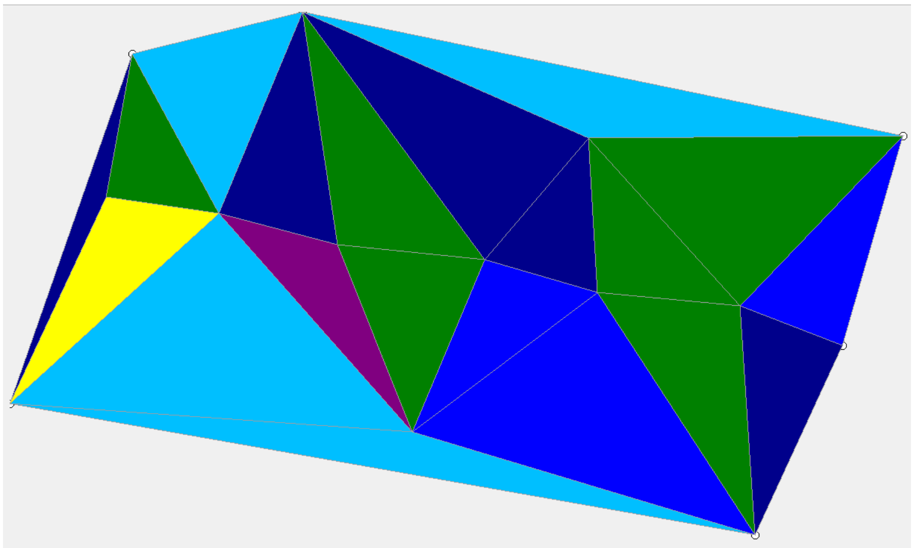


Obrázek 7: Ukázka lineární interpolace kupy

```
# Aspect visualization by cardinal direction
if 25 < A <= 65:
    colors.append(QColor(255, 165, 0))
elif 65 < A <= 105:
    colors.append(QColor(255, 255, 0))
elif 105 < A <= 145:
    colors.append(QColor(0, 128, 0))
elif 145 < A <= 185:
    colors.append(QColor(0, 191, 255))
elif 185 < A <= 225:
    colors.append(QColor(0, 0, 255))
elif 225 < A <= 265:
    colors.append(QColor(0, 0, 139))
elif 265 < A <= 305:
    colors.append(QColor(128, 0, 128))
elif 305 < A <= 345:
    colors.append(QColor(138, 43, 226))
else:
    colors.append(QColor(255, 0, 0))
```

Obrázek 8: Intervaly azimutů a jejich RGB kódy

V neposlední řadě byl program testován i na schopnost správně vykreslovat terénní hrany. V programu ArcGIS Pro proto byl pomocí bodů vytvořen jednoduchý model terénní hrany. Výsledek však není vůbec uspokojivý. Vzhledem k tomu, že triangulace pracuje pouze s 2D daty, není terénní hrana (správně vede středem území ve směru přibližně východ - západ) vůbec zřejmá. Body jsou totiž zpracovávány pouze v rovinách x a y.



Obrázek 9: Model terénní hrany

7 Závěr

Námi vytvořený program splňuje zadání neboť úspěšně z bodů načtených v textovém souboru (popř. ručně naklikaných bodů) generuje polyedrický model terénu a v případě zájmu uživatele je schopen nad tímto modelem vytvořit vrstevnice za pomocí lineární metody a také analyzovat sklon a orientaci trojúhelníků obsažených v modelu.

Program jako celek nemá v zásadě žádné stěžejní nedostatky a všechny své podkroky provádí v rámci možností úspěšně a bez závad. Naprosto nezpochybnitelnou výhodou by však byla důkladnější analýza vytvořeného programu, zejména pak na reálných datech. Tato data lze totiž poté lépe porovnat s realitou, a věrohodněji tak určit skutečnou přesnost sestaveného programu.

Další vylepšení programu nabízí splnění bonusových úloh. Jedná se například o tirangulací nekonvexních oblastí, generování automatického popisu vrstevnic, vytvoření algoritmu pro automatické generování různých terénních tvarů či 3D vizualizace terénu s promítáním.

Při hodnocení vytvořeného digitálního modelu terénu (založeného na 2D Deleaunayho triangulaci) z hlediska kartografického je nutné podotknout, že výsledky nejsou příliš uspokojivé. Zejména co se týká interpretace výsledků, nenabízí nám 2D pohled zrovna jednoduchou úlohu. Třetí dimenze je u takovýchto modelů obzvláště potřebná, a proto je tedy pro uživatele těchto výstupu v mnoha případech velmi těžké rozpoznat, zdali se například jedná o údolí, hřbet, kupu atp. Pokud je pak terén obzvláště členitý, je orientace v modelu takřka nemožná. Pomocnou ruku zde naštěstí podává vykreslení vrstevnic. Ani to si však neporadí se singulárními případy, jakým je například stěna, která by však ve 3D modelu nebyla žádným problémem.

8 Seznam literatury

Zdrojový kód programu vznikl z velké části na základě informací z přednášky a cvičení vedených doc. Bayerem (2022). Informace byly čerpány z prezenčních lekcí a zároveň z prezentace dostupné na odkazu https://web.natur.cuni.cz/ bayertom/images/courses/Adk/adk5_new.pdf (cit. 8. 5. 2022)

LEIFER, F. (2006): Delaunayho triangulace a její aplikace. Diplomová práce. VUT v Brně, FSI ústav automatizace a informatiky