

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

<b>1</b>	<b>Zadání</b>	<b>2</b>
1.1	Údaje o bonusových úlohách . . . . .	3
<b>2</b>	<b>Popis a rozbor problému</b>	<b>3</b>
<b>3</b>	<b>Popis použitých algoritmů</b>	<b>4</b>
3.1	Delaunayho triangulace . . . . .	5
3.1.1	Vlastnosti Delaunayho triangulace . . . . .	5
3.1.2	Implementace metody . . . . .	5
3.2	Vrstevnice . . . . .	6
3.2.1	Implementace metody . . . . .	6
3.3	Sklon terénu . . . . .	7
3.3.1	Implementace metody . . . . .	7
3.4	Expozice terénu . . . . .	7
3.4.1	Implementace metody . . . . .	7
<b>4</b>	<b>Informace o bonusových úlohách</b>	<b>8</b>
4.1	Výběr barevných stupnic při vizualizaci sklonu a expozice . . . . .	8
4.2	Algoritmus pro automatické generování terénních tvarů . . . . .	8
<b>5</b>	<b>Vstupní data</b>	<b>8</b>
<b>6</b>	<b>Výstupní data</b>	<b>8</b>
<b>7</b>	<b>Aplikace</b>	<b>8</b>
<b>8</b>	<b>Dokumentace</b>	<b>8</b>
8.1	Třídy . . . . .	8
8.1.1	Algorithms . . . . .	8
8.1.2	Draw . . . . .	10
8.1.3	SortByXAsc . . . . .	11
8.1.4	Edge . . . . .	11
8.1.5	QPoint3D . . . . .	11
8.1.6	Triangle . . . . .	12
8.1.7	Widget . . . . .	12
<b>9</b>	<b>Závěr</b>	<b>13</b>
9.1	Delaunayho triangulace . . . . .	13
9.2	Vrstevnice . . . . .	13
9.3	Sklon a expozice terénu . . . . .	13
9.4	Výstupy aplikace . . . . .	14
<b>10</b>	<b>Náměty na vylepšení</b>	<b>14</b>
10.1	Sklon terénu . . . . .	14
10.2	Export výsledku . . . . .	14
<b>11</b>	<b>Reference</b>	<b>15</b>

# 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
<b>Max celkem:</b>	<b>65b</b>

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

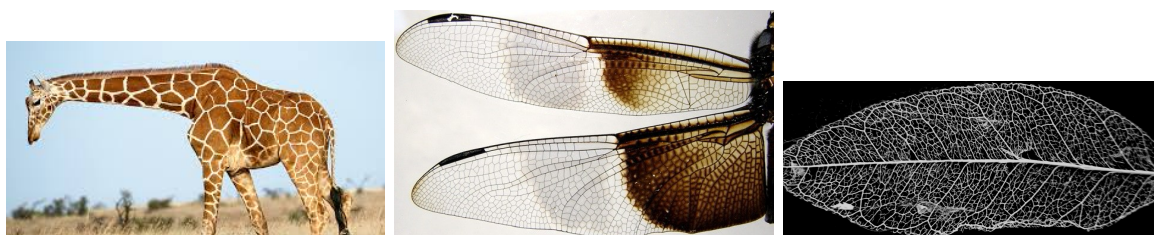
Hlavním cílem této úlohy je tvorba aplikace, která nad výškopisnými body vytvoří Delaunayho triangulaci, vrstevnice a vypočte sklon a expozici k světovým stranám.

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. Delaunayho triangulace tyto podmínky obecně splňuje, přesto existují výjimky, kdy nemá Delaunayho triangulace jednoznačné řešení. Tento stav může nastat pro určité množiny dat, např. pravoúhlý grid. [Zdroj: 2]

Delaunayho 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. Voroniovy diagramy se vyskytují i v přírodě, můžeme je nalézt na kůži žirafy, na křídle vážky a na žilkování listu. [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 Delaunayho 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 nazývány Delaunayho rastry (The Delaunay Raster). Autoři takovýchto děl jsou schopni pomocí Delaunayho 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, Delaunayho 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 Delaunayho triangulací pomocí metody inkrementální konstrukce. Další metody přímé konstrukce Delaunayho triangulace jsou např. metoda

lokálního 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 Delaunayho triangulace

Delaunayho triangulace je nejčastěji používanou triangulací při tvorbě digitálního modelu terénu. Delaunayho 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 Delaunayho 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 Delaunayho bodu:  $\underline{p} = \operatorname{argmin}_{\forall p_i \in \sigma_L(e)} r'(k_i); k_i = (a, b, p_i); e = (a, b)$   
Při nenalezení Delaunayho 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 Delaunayho bodu k hraně z AEL (viz 4).

Pokud Delaunayho 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 pětinásobek 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:

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

$$v_x = \Delta x_2, x_3; u_y = \Delta y_2, y_3; u_z = \Delta z_2, z_3;$$

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:

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

$$v_x = \Delta x_2, x_3; u_y = \Delta y_2, y_3; u_z = \Delta z_2, z_3;$$

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 a grid. Přesto, že grid neoznačuje žádný terénní tvar, byl mezi možnostmi přidán z toho důvodu, abychom demonstrovali špatnou implementaci takového rozmístění bodů.

## 5 Vstupní data

V aplikaci lze buď tvořit body pomocí kurzoru, kde však je výška generována náhodně, nebo lze generovat jednotlivé útvary terénu a nebo lze importovat data ve formátu .txt

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 Delaunayho triangulaci, se zvoleným krokem vykreslí vrstevnice a následně je možné zobrazit sklon či expozici terénu.

## 7 Aplikace

## 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 Delaunayho vlastnosti.

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

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

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

Metoda vypočte průsečík hrany, tvořené 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  $dz$ )**

Metoda z triangulace dt v zadaném intervalu v ose Z  $\langle z_{min}; z_{max} \rangle$  s intervalem vrstevnic dz vrátí vektor hran definující vrstevnice.

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

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

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

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

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

**std::vector<QPoint3D>generateHill()**

**std::vector<QPoint3D>generateValley()**

**std::vector<QPoint3D>generateMountains()**

**std::vector<QPoint3D>generateRest()**

### 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 Delaunayho triangulace z vykreslovacího okna.

**void setDT(std::vector<Edge>&dt\_)**

Metoda slouží pro převod Delaunayho triangulace do vykreslovacího okna.

**void setContours(std::vector<Edge>&contours\_)**

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 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.

## 9 Závěr

Byla vytvořena aplikace, která nad importovanými body vytvoří Delaunayho triangulaci, vykreslí vrstevnice, sklon a expozici terénu.

### 9.1 Delaunayho 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 selhávají. Obecně se ví, že Delaunayho triangulace nemá pro body na mřížce jednoznačné řešení. Tento problém by se mohl 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. Důvod je zřejmý z implementace triangulace.

V aplikaci též nelze nastavit zájmovou oblast, proto je nutné před výpočtem algoritmu odstranit body, nad kterými nechceme provádět triangulaci.

Aplikace je užitečná pro výškopisná data v terénu bez povinných hran a kde data nepotřebují být upravována, viz předchozí odstavec.

### 9.2 Vrstevnice

Pro vykreslení vrstevnic byla použita lineární interpolace, která má svoje 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.

Autoři neřešili vykreslení hlavní vrstevnice zesílenou čarou ani popis vrstevnic.

### 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 jednotlivé 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 určité přecházení 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.

Do aplikace by se hodila legenda expozice terénu.

## 9.4 Výstupy aplikace

Uživatel má v aplikaci možnost uložit si vytvořený model naimportovaných souřadnic, Delaunayho triangulací, vrstevnic, sklonu či expozice.

# 10 Náměty na vylepšení

V aplikaci by se dalo vylepšit zobrazení vrstevnic, mohla by být zvýrazněná zesílená vrstevnice a zobrazen popis vrstevnic. V aplikaci mi mohlo být řešeno odstranění naimportovaných bodů, nad kterými uživatel nechce počítat triangulaci. V aplikaci by mohla být volba pro uživatele nastavení výpočet vrstevnic v určitém rozsahu nadmořské výšky.

## 10.1 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, by jí následně mohli využívat například stavaři, kteří při výkopu musí zkontrolovat sklon vzhledem k druhu horniny a přípustnému sklonu svahu.

## 10.2 Export výsledku

Při exportu vzniklého rastru je v aplikaci možnost výsledek uložit. Není však ošetřena cesta místa uložení. V současné verzi je obrázek ukládán v rámci složky "Release" či "Debug", v závislosti na tom, v jakém režimu uživatel aplikaci spustil.

## 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](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>