

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ, OBOR GEODÉZIE, KARTOGRAFIE A GEOINFORMATIKA
KATEDRA SPECIÁLNÍ GEODÉZIE

název předmětu

ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

název úlohy

Úloha 3: Digitální model terénu a jeho analýzy

akademický
rok

2023/24

semestr

Letní

vypracoval(a)

Jan Koudelka, Vojtěch Müller

Datum

22.5.2024

klasifikace

Technická zpráva

1 Zadání

Metodou inkrementální konstrukce vytvořte nad množinou P vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů). Nad 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.

Vstupní množinu bodů včetně výše uvedených bodů vhodně vizualizujte.

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

2 Kroky

V rámci úlohy byly zpracovány tyto části:

- Delaunay triangulace a polyedrický model terénu. – 10 bodů
- Konstrukce vrstevnic a analýza sklonu a expozice. – 10 bodů

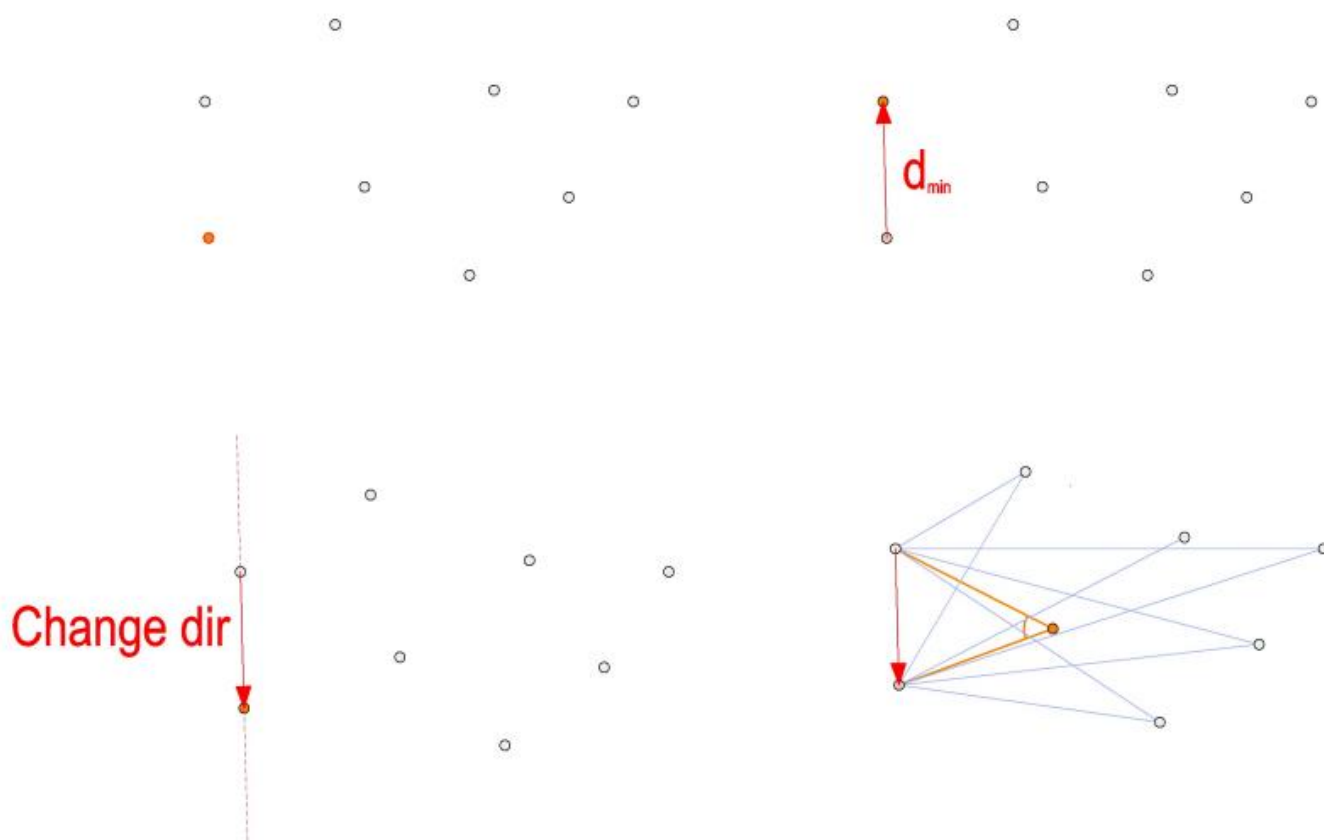
3 Algoritmy

Úloha se zabývá vizualizací digitálního modelu terénu a analýze na tomto modelu. Pro výstup byla vytvořena jednoduchá aplikace, ve které jsou data zobrazována. Do této aplikace byly zpracovány následující algoritmy.

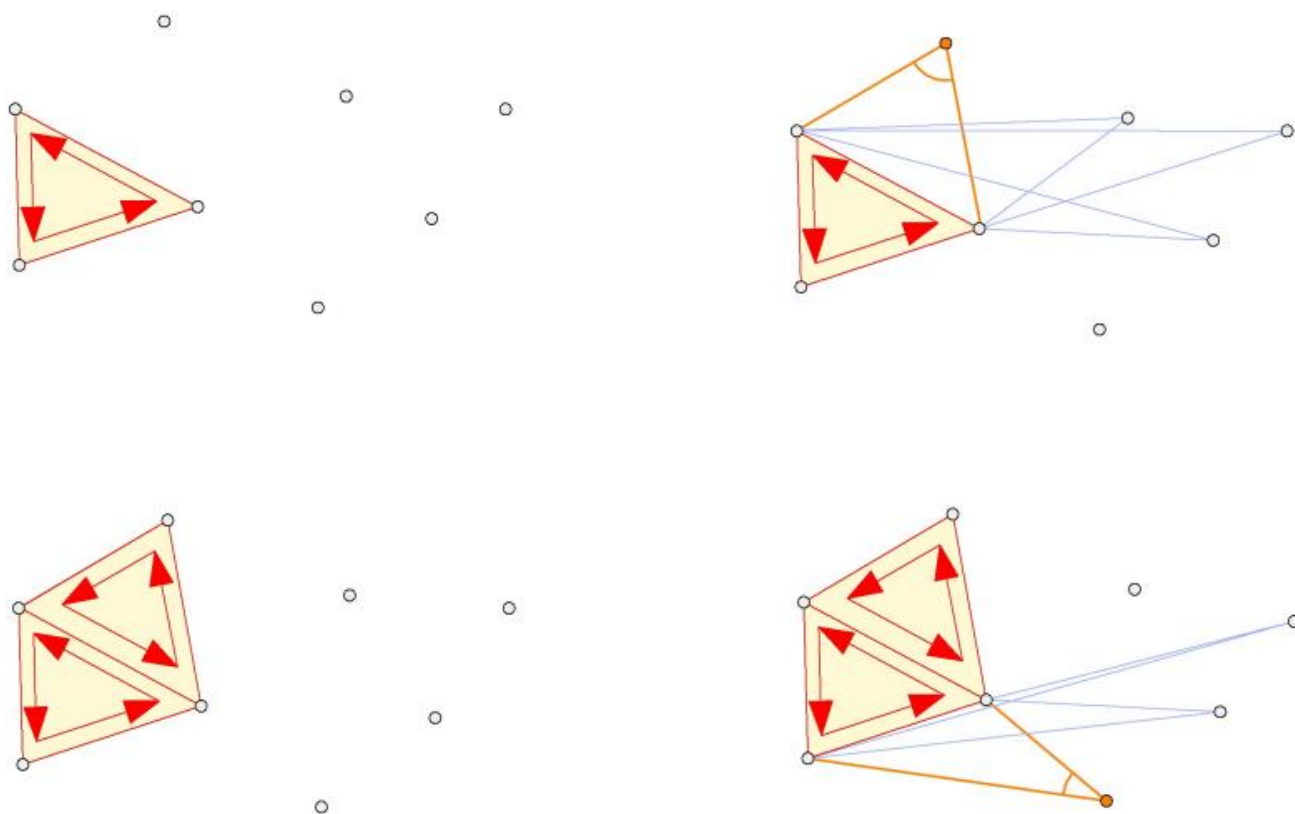
3.1 Delaunay triangulace

Pomocí Delaunay triangulace bude vytvořena síť ze zadaných bodů. Na této síti poté pracují všechny ostatní algoritmy. Pro tuto úlohu je zpracován algoritmus pro metodu inkrementální konstrukce.

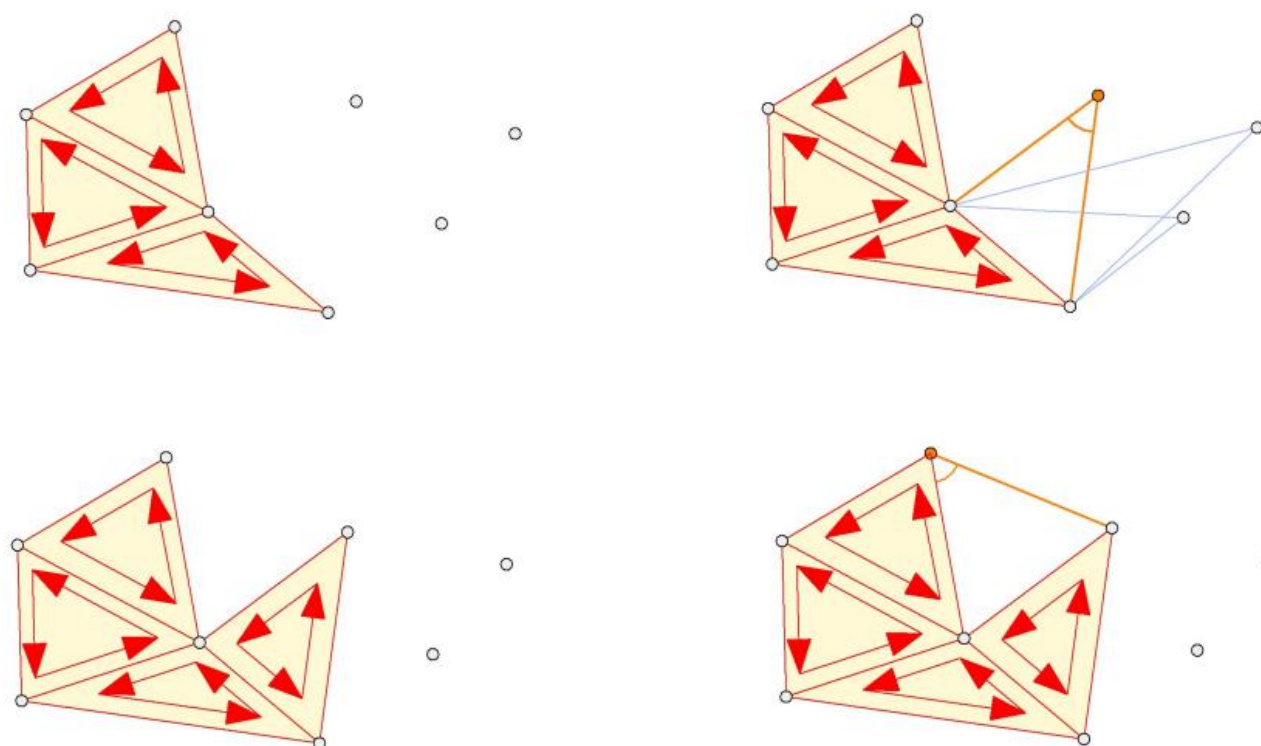
Metoda je založena na postupném přidávání bodů do již vytvořené triangulace. K určité hraně je vždy dohledáván vhodný bod, který splňuje podmínku, že úhel u tohoto bodu mezi body prohledávané hrany musí být maximální. Takto vyhledaný bod tvoří nový trojúhelník. Při konstrukci algoritmu je využíván Active Edge List (AEL), ve kterém se ukládají hrany, pro které hledáme vhodný bod.



Obrázek 1: Postup metody inkrementální konstrukce 1/3 [1]



Obrázek 2: Postup metody inkrementální konstrukce 2/3 [1]



Obrázek 3: Postup metody inkrementální konstrukce 3/3 [1]

Algoritmus 1: Metoda inkrementální konstrukce

```
def createDT(self, points:list[QPoint3DF]):
    #return list of edges
    dt = []
    #active edges list
    ael = []

    #find start point
    p1 = min(points, key = lambda k: k.x())

    #find nearest point
    min_dis, min_i = self.getNearestPoint(p1, points)
    p2 = points[min_i]

    #create edges
    e = Edge(p1,p2)
    e_op = Edge(p2,p1)

    #add to AEL
    ael.append(e)
    ael.append(e_op)

    #repeat until AEL is empty
    while ael:
        #take the first edge
        e1 = ael.pop()

        #switch orientation
        e1_op = e1.changeOrientation()

        #find optimal delaunay point
        omega_max, i_max = self.getDelauneyPoint(e1_op.getStart(), e1_op.getEnd(),
points)

        #is there any delaunay point
        if i_max >= 0:

            #create remaining edges
            e2 = Edge(e1_op.getEnd(), points[i_max])
            e3 = Edge(points[i_max], e1_op.getStart())

            #add triangle to DT
            dt.append(e1_op)
            dt.append(e2)
            dt.append(e3)

            self.updateAEL(e2, ael)
            self.updateAEL(e3, ael)

    return dt
```

V algoritmu jsou nejprve vytvořeny seznamy trojúhelníků a aktivních hran (AEL). Následně je nalezen výchozí bod, který je volen jako bod s minimální souřadnicí X, a poté je nalezen bod, který je k tomuto bodu nejbližší. Tyto dva body poté tvoří základní hranu. Tato hrana a také hrana opačně orientovaná jsou poté přidány do AEL. Dále je ve while cyklu procházen AEL, z něj jsou postupně odebírány hrany a na nich je provedeno vytváření trojúhelníků. Pro každou hranu je nejprve obrácena rotace a následně je nalezen vhodný bod z těch, které ještě nebyly pro triangulaci použity. Pokud je takový bod nalezen, pak jsou vytvořeny hrany k tomuto bodu a do seznamu trojúhelníků je přidán nalezený trojúhelník. Nakonec musí být aktualizován AEL, tedy vloženy obě nově nalezené hrany a odstraněny opačně orientované hrany, aby již nebyl znovu procházen stejný trojúhelník.

3.2 Vrstevnice

Nad trojúhelníky vytvořenými Delaunay triangulací bude prováděno zobrazení vrstevnic pomocí metody lineární interpolace.

Tato metoda je založena na analytické geometrii. V prohledávaném trojúhelníku hledáme průsečnici s vodorovnou rovinou v určité výšce.

Body průsečíku v trojúhelníku určíme pomocí vzorců

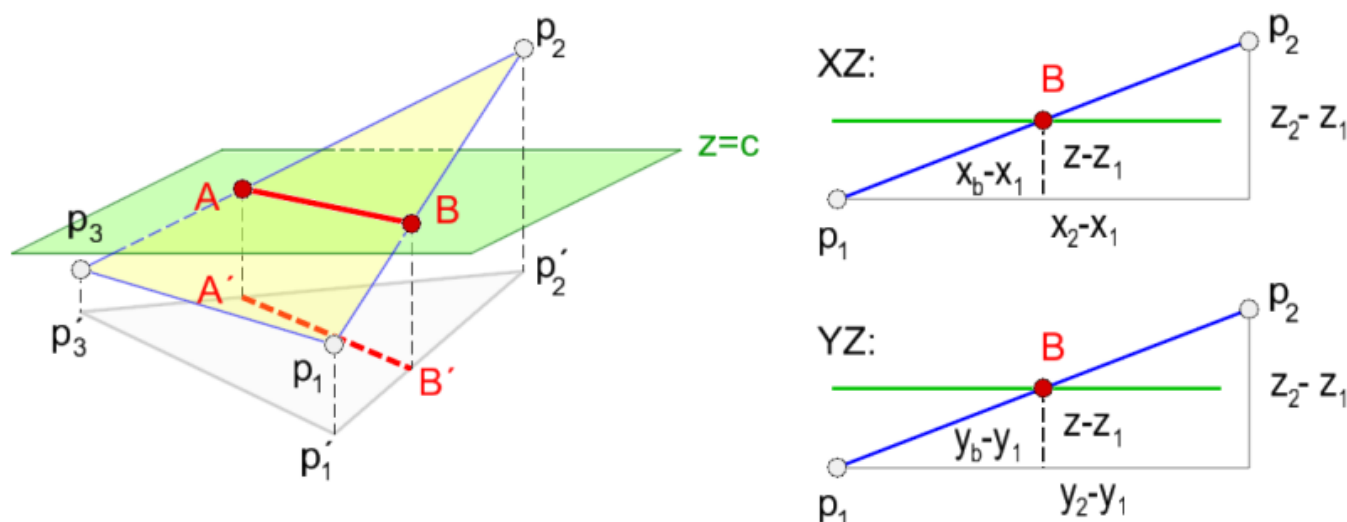
$$x_a = \frac{x_3 - x_1}{z_3 - z_1} (z - z_1) + x_1,$$

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

$$y_a = \frac{y_3 - y_1}{z_3 - z_1} (z - z_1) + y_1,$$

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

které vychází z následujícího obrázku.



Obrázek 4: Konstrukce vrstevnic lineární interpolací [1]

Algoritmus 2: Lineární interpolace vrstevnic

```
def createContourLine(self, dt:list[Edge], zmin:float, zmax:float, dz:float):  
    #create contour lines with max z, min z and interval z  
    contours = []  
  
    #process all triangles  
    for i in range(0, len(dt),3):  
        #get triangle vertices  
        p1 = dt[i].getStart()  
        p2 = dt[i].getEnd()  
        p3 = dt[i+1].getEnd()  
  
        #get Z of points  
        z1 = p1.getZ()  
        z2 = p2.getZ()  
        z3 = p3.getZ()  
  
        #test horizontal plane and triangle intersections  
        for z in arange(zmin, zmax, dz):  
            dz1 = z-z1  
            dz2 = z-z2  
            dz3 = z-z3  
  
            #triangle is coplanar  
            if (dz1==0 and dz2==0 and dz3==0):  
                continue  
  
            #edge 12 in plane  
            elif(dz1==0 and dz2==0):  
                contours.append(dt[i])  
  
            #edge 23 in plane  
            elif(dz2==0 and dz3==0):  
                contours.append(dt[i+1])  
  
            #edge 31 in plane  
            elif(dz3==0 and dz1==0):  
                contours.append(dt[i+2])  
  
            #edges 12 and 23 intersected by plane  
            elif(dz1*dz2<=0 and dz2*dz3<0 or dz1*dz2<0 and dz2*dz3<=0):  
                #compute intersections  
                a = self.getContourPoint(p1,p2,z)  
                b = self.getContourPoint(p2,p3,z)  
  
                #create edge and append to the list  
                e = Edge(a,b)  
                contours.append(e)  
  
            #edges 23 and 31 intersected by plane  
            elif(dz2*dz3<=0 and dz3*dz1<0 or dz2*dz3<0 and dz3*dz1<=0):
```

```

#compute intersections
a = self.getContourPoint(p2,p3,z)
b = self.getContourPoint(p3,p1,z)

#create edge and append to the list
e = Edge(a,b)
contours.append(e)

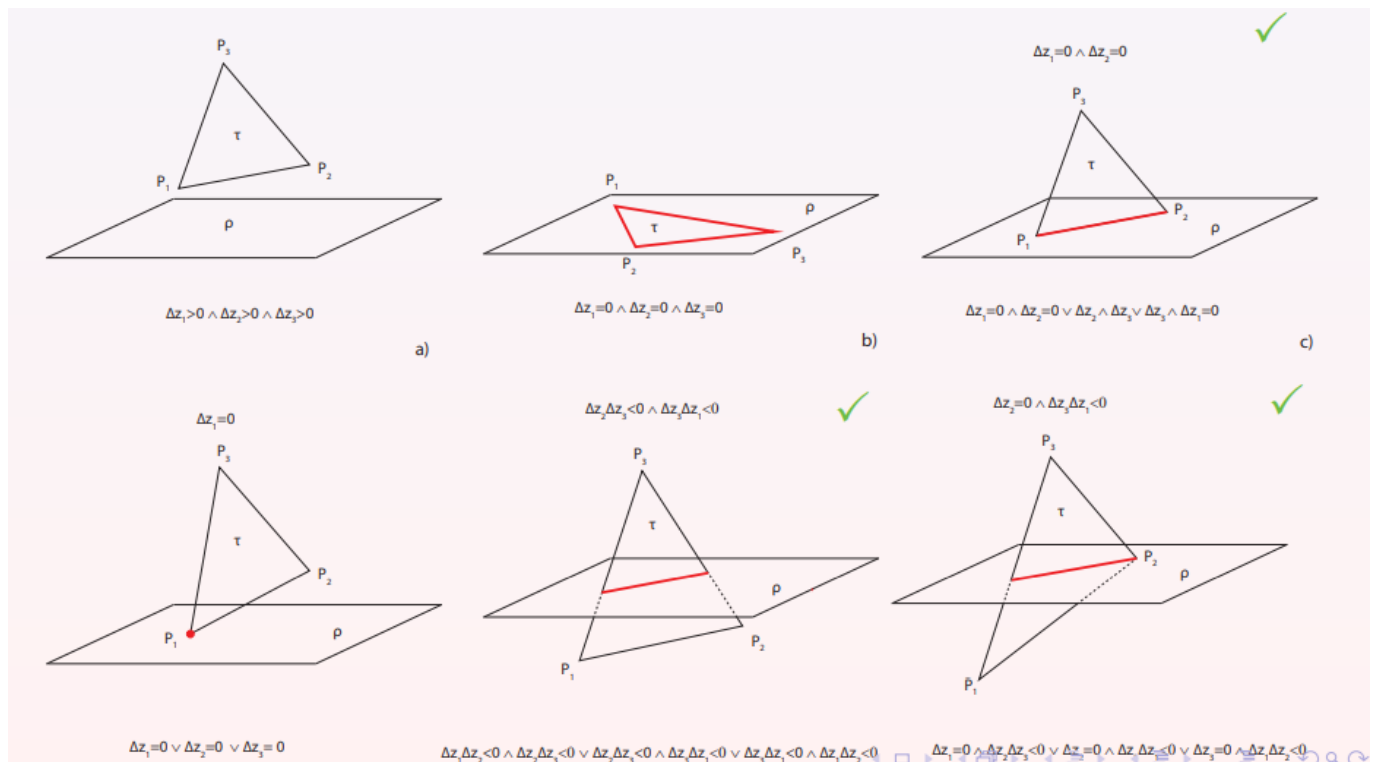
#edges 31 and 12 intersected by plane
elif(dz3*dz1<=0 and dz1*dz2<0 or dz3*dz1<0 and dz1*dz2<=0):
    #compute intersections
    a = self.getContourPoint(p3,p1,z)
    b = self.getContourPoint(p1,p2,z)

    #create edge and append to the list
    e = Edge(a,b)
    contours.append(e)

return contours

```

V algoritmu je nejprve vytvořen seznam vrstevnic. Následně je for cyklem procházen každý trojúhelník vstupující triangulace. Pro každý trojúhelník jsou hledány průsečíky pro jednotlivé výšky. V algoritmu jsou vloženy podmínky, které řeší, který z případů v následujícím obrázku nastal.



Obrázek 5: Možnosti průsečíku trojúhelníku a vodorovné roviny [1]

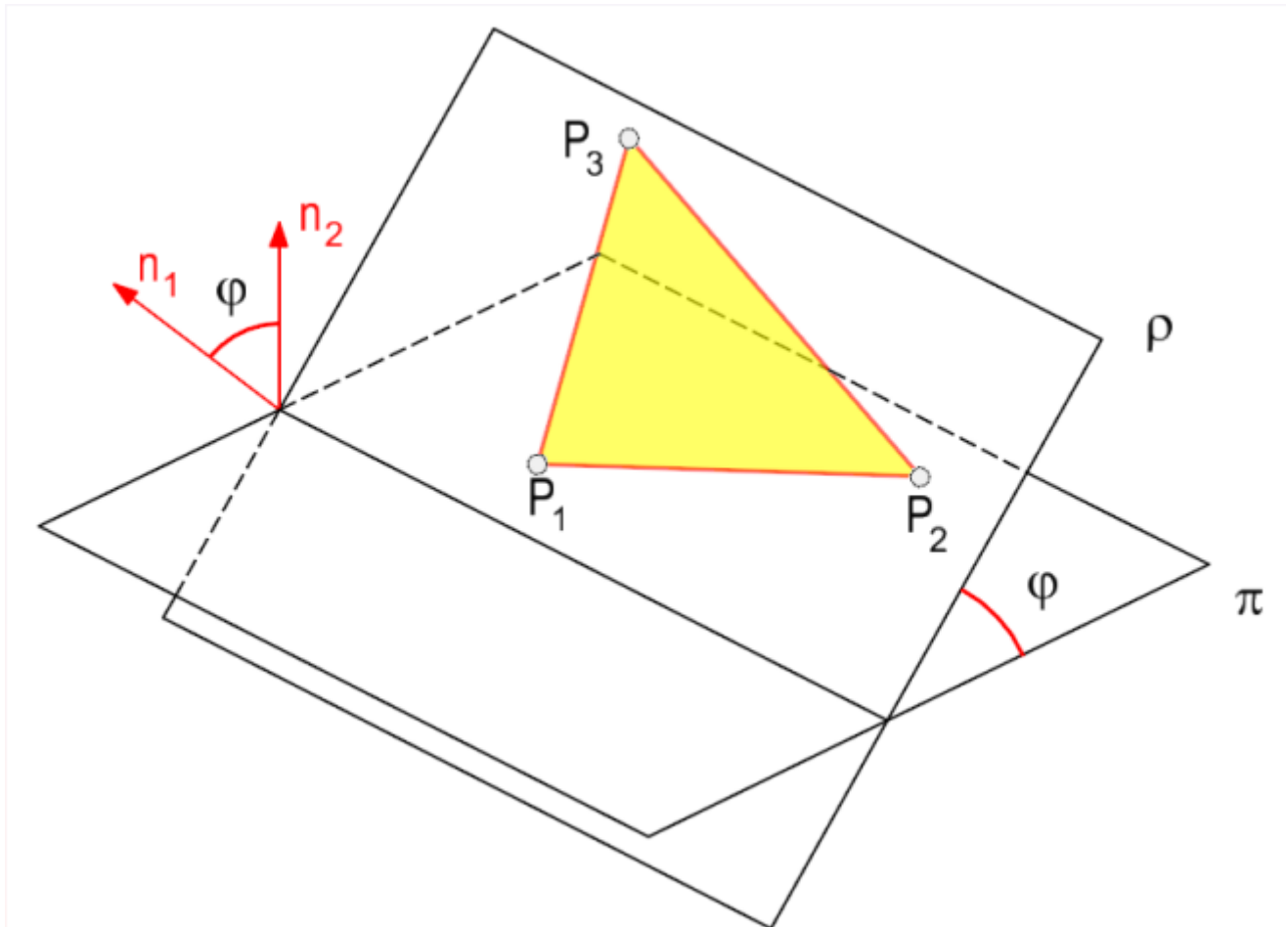
Podle toho, jaký z případů nastal, je pak do seznamu souřadnic přidána hrana, která představuje vrstevnici.

3.3 Analýza sklonu digitálního modelu terénu

Analýza probíhá v jednotlivých trojúhelnících, kdy je počítán úhel mezi normálovým vektorem vodorovné roviny a normálovým vektorem daného trojúhelníka. Tento úhel je počítán ze vzorce

$$\varphi = \arccos \frac{n_1 \cdot n_2}{\|n_1\| \|n_2\|} = \frac{n_{1z}}{\|n_1\|},$$

kde $n_1 = (a, b, c)$ a $n_2 = (0, 0, 1)$.



Obrázek 6: Výpočet sklonu terénu [1]

Algoritmus 3: Výpočet sklonu terénu

```
def computeSlope (self, p1:QPoint3DF, p2:QPoint3DF, p3:QPoint3DF):  
    # compute triangle slope  
  
    #vectors  
    ux = p1.x()-p2.x()  
    uy = p1.y()-p2.y()  
    uz = p1.getZ()-p2.getZ()  
  
    vx = p3.x()-p2.x()  
    vy = p3.y()-p2.y()  
    vz = p3.getZ()-p2.getZ()  
  
    #normal vector  
    nx = uy*vz-uz*vy
```

```

ny = uz*vx-ux*vz
nz = ux*vy-uy*vx

#norm of vector
norm = sqrt(nx*nx + ny*ny + nz*nz)

#slope
cosfi = abs(nz)/norm
return acos(cosfi)

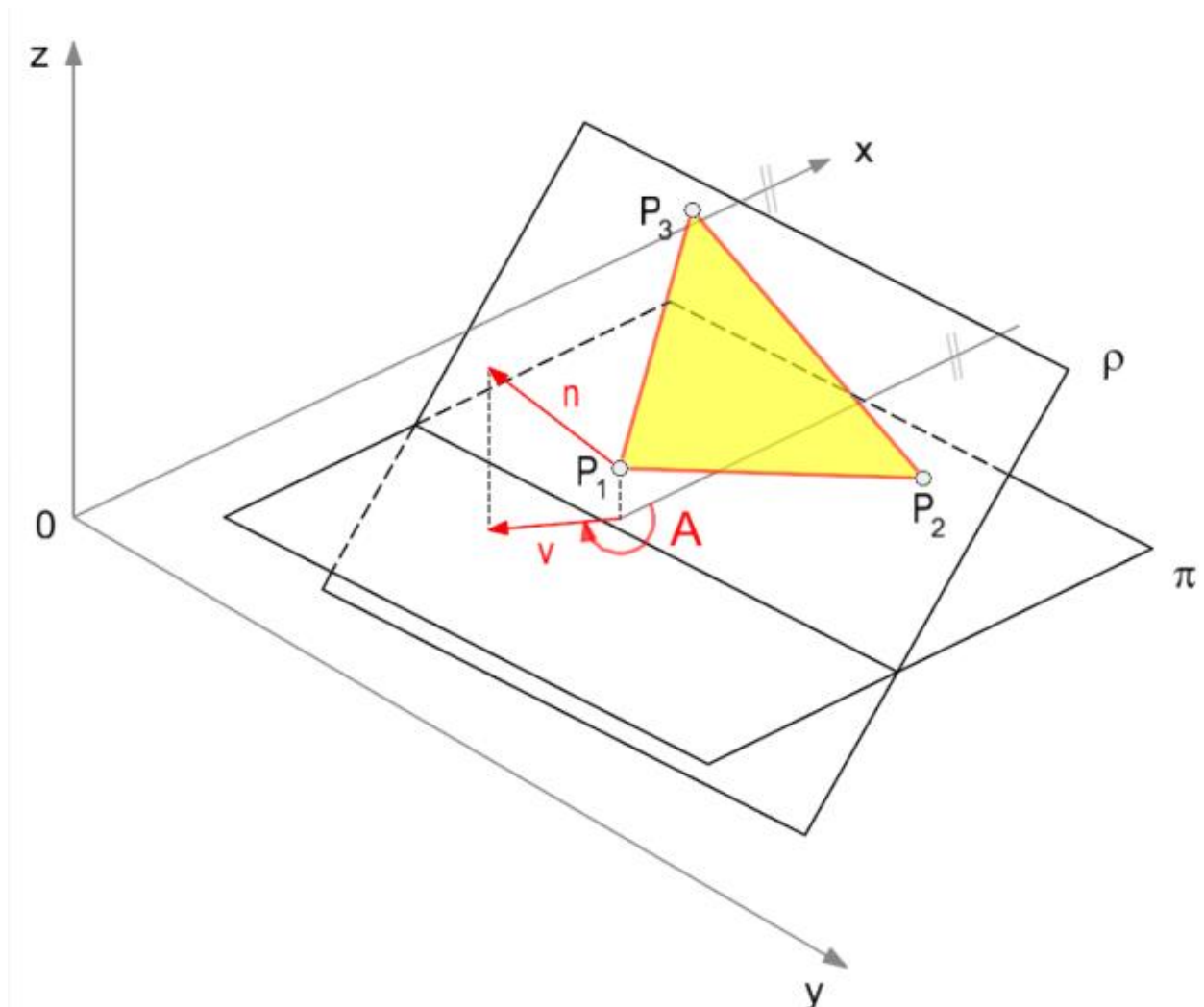
```

Sklon je následně vizualizován pro každý trojúhelník ve stupních šedi podle velikosti úhlu φ .

3.4 Analýza expozice digitálního modelu terénu

Tato analýza také probíhá pro jednotlivé trojúhelníky. Počítán je úhel od severu po směru hodinových ručiček ve vodorovné rovině. Tento úhel je počítán k průmětu normálového vektoru daného trojúhelníka. Výpočet probíhá ze vzorce

$$A = \text{atan} \frac{n_{1x}}{n_{1y}}.$$

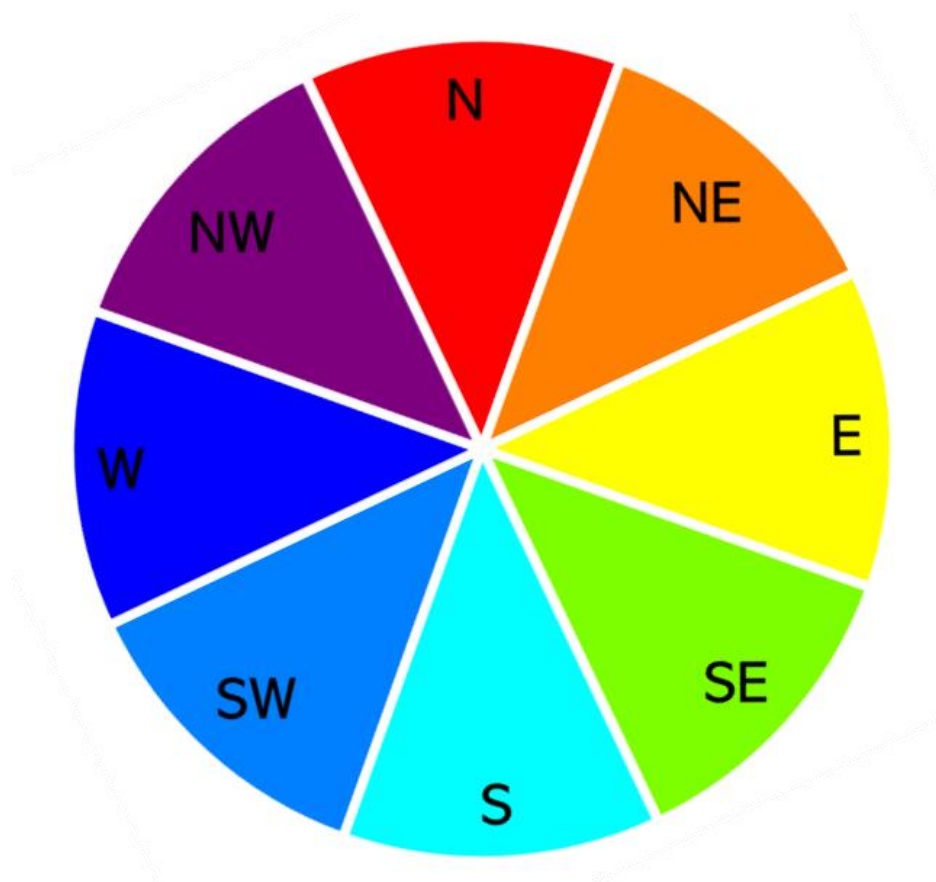


Obrázek 7: Výpočet orientace terénu [1]

Algoritmus 4: Výpočet orientace terénu

```
def computeAspect(self, p1:QPoint3DF, p2:QPoint3DF, p3:QPoint3DF):  
    # compute triangle aspect  
  
    #vectors  
    ux = p1.x()-p2.x()  
    uy = p1.y()-p2.y()  
    uz = p1.getZ()-p2.getZ()  
  
    vx = p3.x()-p2.x()  
    vy = p3.y()-p2.y()  
    vz = p3.getZ()-p2.getZ()  
  
    #normal vector  
    nx = uy*vz-uz*vy  
    ny = uz*vx-ux*vz  
  
    #aspect  
    aspect_help = atan2(nx,ny)  
    if (aspect_help<0):  
        aspect_help = aspect_help+2*pi  
    return aspect_help
```

Orientace terénu je vizualizována pro jednotlivé trojúhelníky podle barevné stupnice, která znázorňuje světovou stranu, na kterou je sklon orientován. Legendu této stupnice lze pozorovat v následujícím obrázku.



Obrázek 8: Legenda expozice terénu

4 Problematické části

Základní vytvoření aplikace probíhalo bez větších komplikací. Vytvoření prostředí, které zde probíhalo bylo obdobné jako v první a druhé úloze. Největší náročnost úlohy probíhala ve zpracování samotných algoritmů.

Samotné vykreslení vrstevnic nepředstavovalo komplikace, ovšem při nadstavbových funkcích určité problémy vznikaly. Například při nastavení intervalu určovaných vrstevnic a odstupu jednotlivých vrstevnic bylo potřeba vyřešit problém, protože propojení nejprve neprobíhalo. Tento problém byl ale nakonec vyřešen.

Jednou z komplikovanějších částí bylo také vykreslení stupnice pro orientaci terénu, která prochází skrze tři barevné kanály.

5 Vstupní data

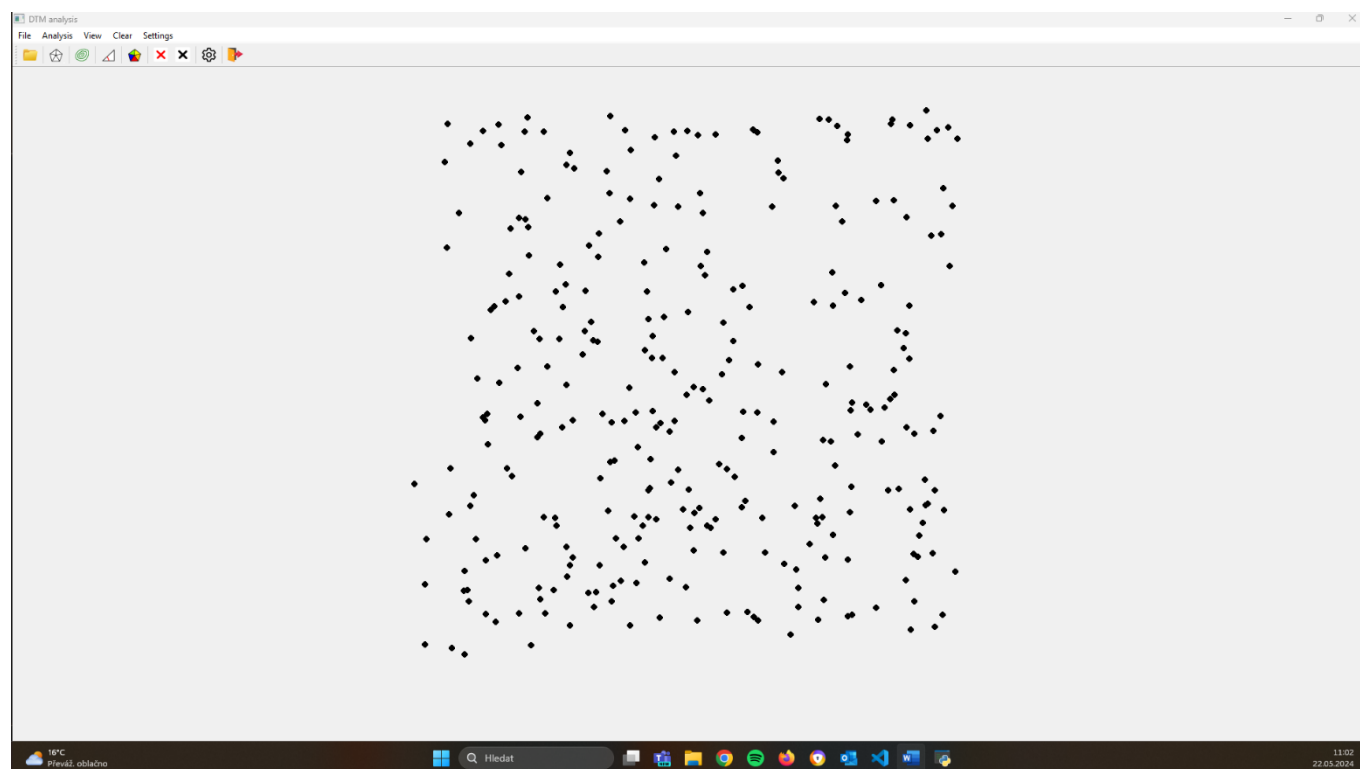
Pro vstup do aplikace byl vytvořen shapefile soubor. Jedná se o soubor s body, který byl vytvořen z dat DMR5G. Tento soubor byl vytvořen v softwaru ArcGIS Pro a body v něm jsou body v okolí vrcholu kopce Pecný u Ondřejova.

6 Výstupní data

Výstupem úlohy je aplikace, která umožní nahrát shapefile soubor s body a na tomto souboru umožňují provést zobrazení Delaunay triangulace, vrstevnic, sklonu terénu a orientace terénu.

Výstupem z úlohy je také zhodnocení vhodnosti metod použitých v aplikaci a také zhodnocení výstupů pro jednotlivé druhy terénu. Tento výstup se nachází v kapitole závěr.

7 Výsledná aplikace



Obrázek 9: Výsledná aplikace

8 Dokumentace

Aplikace při otevření zobrazí okno s prostorem pro vstup a výstup a v horní části se nachází lišta s prvky, které je možné použít. Pod lištou se také nachází Toolbar s jednotlivými možnostmi.

Lišta se skládá z pěti částí.

- File
- Analysis
- View
- Clear
- Settings

V části File se poté nachází metoda Open, která umožní nahrát vstup shapefilu do aplikace a metoda Exit, kterou lze aplikaci zavřít.

V části Analysis se poté nachází jednotlivé algoritmy pro výpočet analýzy bodů.

- Create DT – vytvoří triangulaci
- Create contour lines – vytvoří vrstevnice
- Analyse slope – vytvoří analýzu sklonu terénu
- Analyse aspect – vytvoří analýzu orientace terénu

V části View lze zaškrtnout, která z analýz bude ve výstupové části zobrazena a to

- DT
- Contour lines
- Slope
- Aspect

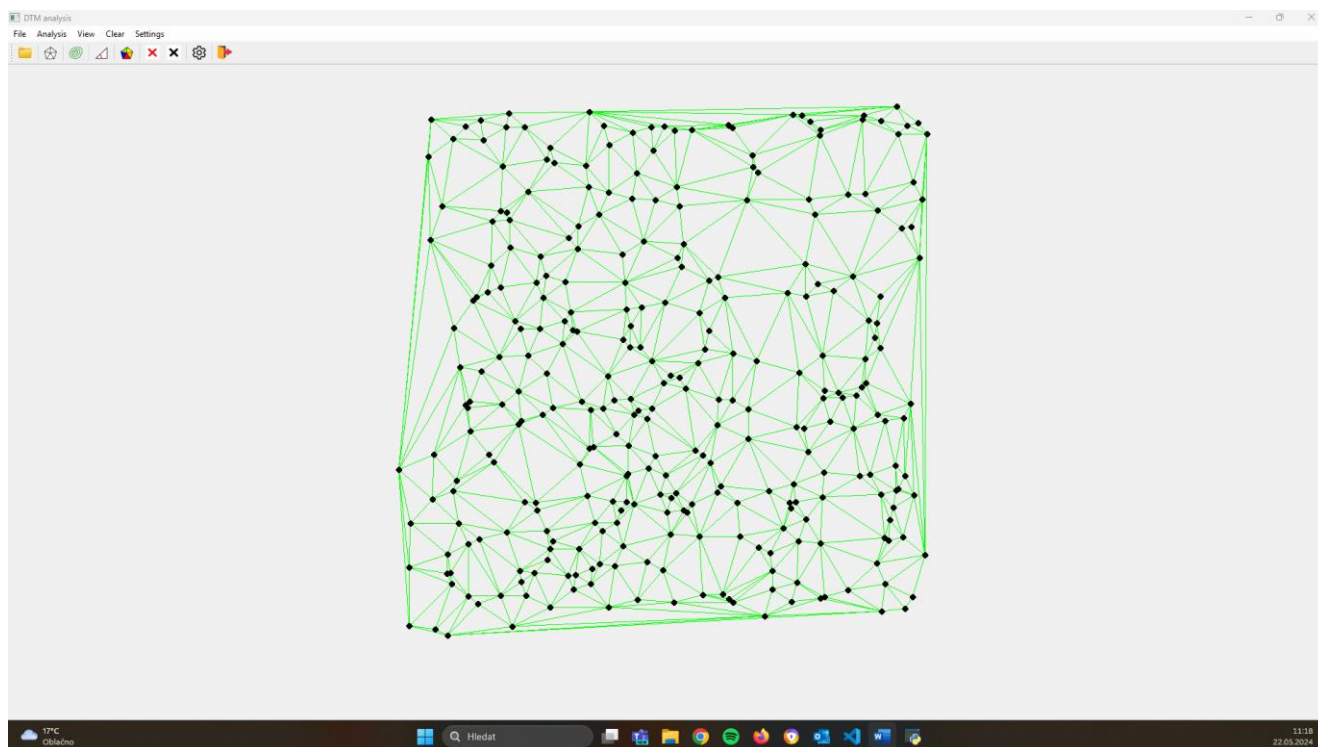
V části Clear lze z pohledové části odstranit výsledky analýz a zachovat body anebo také odstranit vše, včetně vstupu.

V poslední části Settings se při rozkliknutí volby Parameters zobrazí vyskakovací okno, kde lze nastavit hodnoty pro výpočet vrstevnic, konkrétně minimální a maximální hodnotu výšky a interval mezi jednotlivými vrstevnicemi.

9 Závěr

Byla vytvořena aplikace, která zpracovává mračno bodů. Na tomto mračnu lze vytvořit Delauay triangulaci, vrstevnice, analýzu sklonu a analýzu expozice terénu. Veškerá funkčnost aplikace byla testována na bodech vytvořených z DMR 5G v okolí obce Ondřejov.

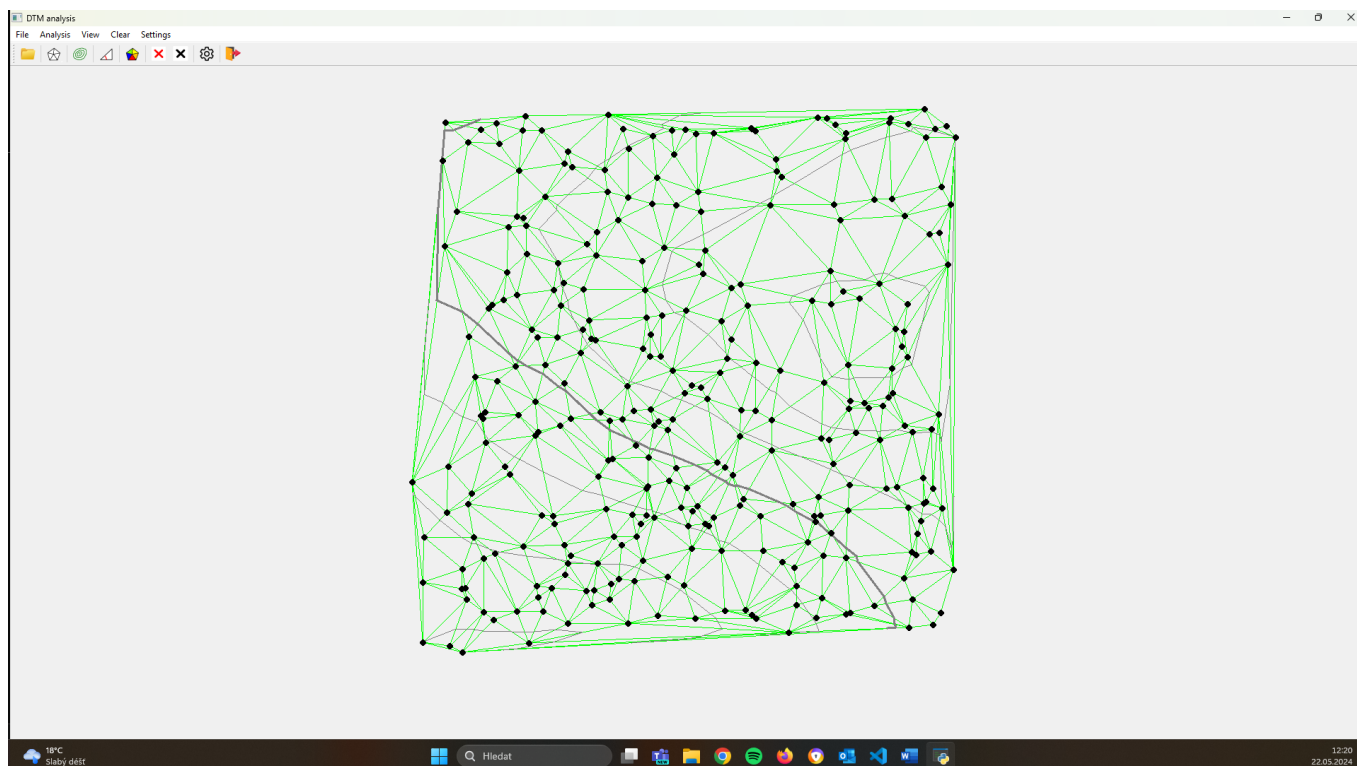
Nad vloženými body je nejprve potřeba vytvořit triangulaci. Tu lze pozorovat na následujícím obrázku.



Obrázek 10: Delaunay triangulace mračna bodů

Jedním z hlavních problémů této triangulace jsou jistě okraje území. V těchto místech vznikají úzké trojúhelníky a další činnosti nad touto triangulací jsou tedy v těchto místech nevhodné. Tato problematika lze řešit více způsoby. Jedním z nich je možnost ručního odstraňování trojúhelníku, aby byly tyto oblasti eliminovány. Dále by také bylo možné eliminovat tyto oblasti různými algoritmy, které by řešily například maximální možné protažení trojúhelníka. Důležitou součástí k správné triangulaci je také rozložení bodů. Zde jsou body rozloženy náhodně a triangulace tedy může v některých oblastech vytvářet nevhodný výstup. Ideálním rozložením je pravidelná čtvercová síť s případným zahuštěním v příkřejších oblastech.

Nad triangulací lze následně vytvářet analýzy, které jsou součástí této aplikace. První z nich je vykreslení vrstevnic, které zobrazují základní tvar terénu. Vrstevnice lze pozorovat v následujícím obrázku. Je také možné nastavit parametry vrstevnic. Konkrétně lze nastavit jejich výškový rozestup, který je v základu nastaven na 10 metrů a také lze nastavit minimální a maximální hodnotu, pro které budou vrstevnice vykreslovány. Pro testovací data byl vybrán interval, pro který jsou zobrazeny všechny vrstevnice.



Obrázek 11: Vrstevnice

Z vrstevnic lze pozorovat, že se v území jedná o vrchol kopce a převážně jeho jihozápadní svah. Také lze vidět, že se vrstevnice v každé páté zvýrazňují.

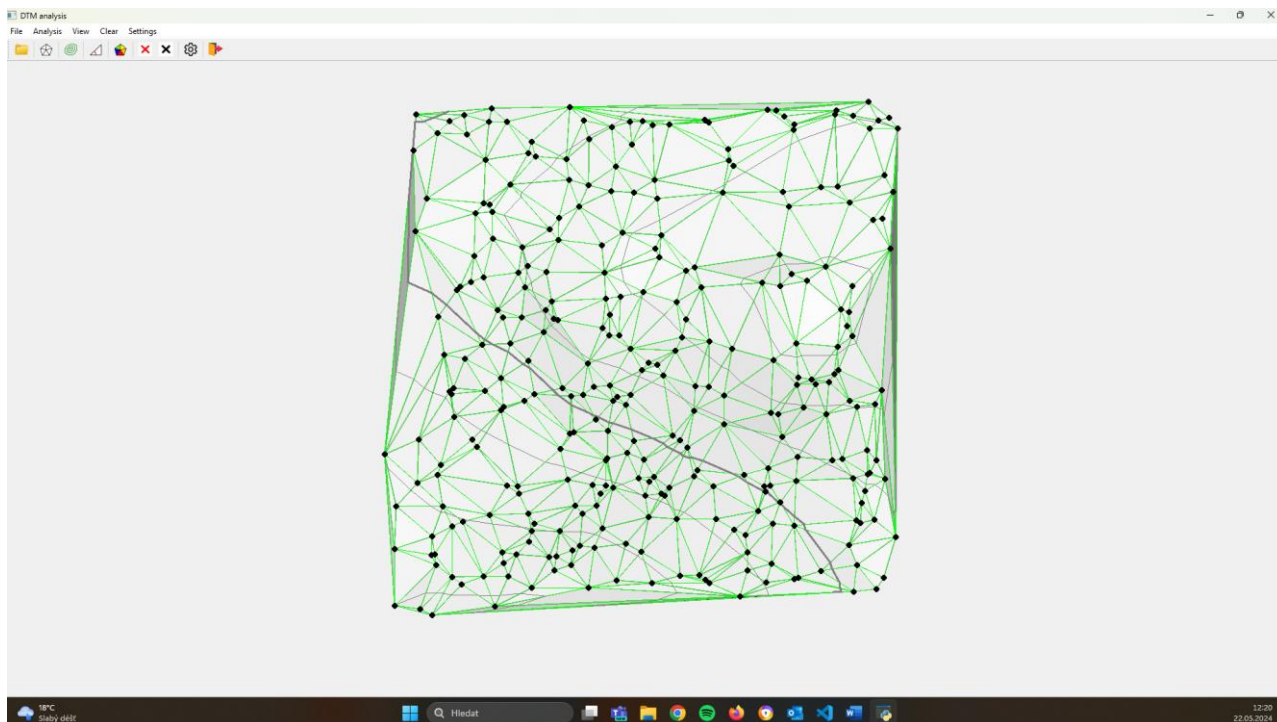
Lze pozorovat problémy, které při vykreslení vznikají. Jedním z nich je právě okrajové území, kde protáhlé trojúhelníky prudce zalamují vrstevnici, jako lze pozorovat na západě území. Při základním vykreslení také vznikají problémy s hladkostí vrstevnic. Vrstevnice se v některých místech příliš prudce lámou. K vylepšení kartografického zobrazení vrstevnic by tedy bylo potřeba použít například některý vyhlazovací algoritmus.

V rovných oblastech s mírnými přechody by také mohly vznikat nechtěné ostrůvky s vrstevnicí. Tyto malé samostatné prvky jsou klasicky také součástí následných úprav, které mohou probíhat ručně, nebo také pomocí algoritmů. Je ovšem nutno počítat s tím, že algoritmy nemusí odvést vždy dokonalou práci a určitá manuální kontrola je často vhodná.

Ke kompletnímu zobrazení vrstevnic je také potřeba uvažovat takzvané povinné hrany. Jedná se o hrany, které klasický algoritmus nenalezne, ale jsou to hrany, které jsou v terénu viditelné. Jedná se převážně o přechody mezi sklonem terénu, například hřbetnice, které se nachází mezi svahy, které klesají na obě strany od této hřbetnice. Dalším prvkem může být údolnice, která naopak zobrazuje místo, od kterého svah roste na obě strany. Tyto hrany vylepší kvalitu zobrazení a určí lepší pravdivost údajů. Problémem těchto hran je ovšem jejich přidávání, které nelze příliš dobře automatizovat a jejich zpracování probíhá ručně.

U zobrazení vrstevnic by také mohl vzniknout problém v příliš prudkém svahu. V tomto území by mohlo být zobrazeno příliš mnoho vrstevnic, které by se poté překrývaly navzájem. Znovu je zde tedy potřeba provést některé činnosti k úpravě. Je vhodné v těchto úsecích vrstevnice nezobrazit a území buďto nechat prázdné, pokud je pochopitelné, že zde vrstevnice pokračují, nebo vrstevnice v tomto území nahradit některou jinou metodou, například některým typem šraf.

Další analýzou, kterou lze nad triangulací zobrazit je sklon terénu, který lze pozorovat na následujícím obrázku. Zobrazení probíhá pomocí stupňů šedi. Více tmavá barva zobrazuje větší sklon terénu a bílá naopak zobrazuje rovný terén.

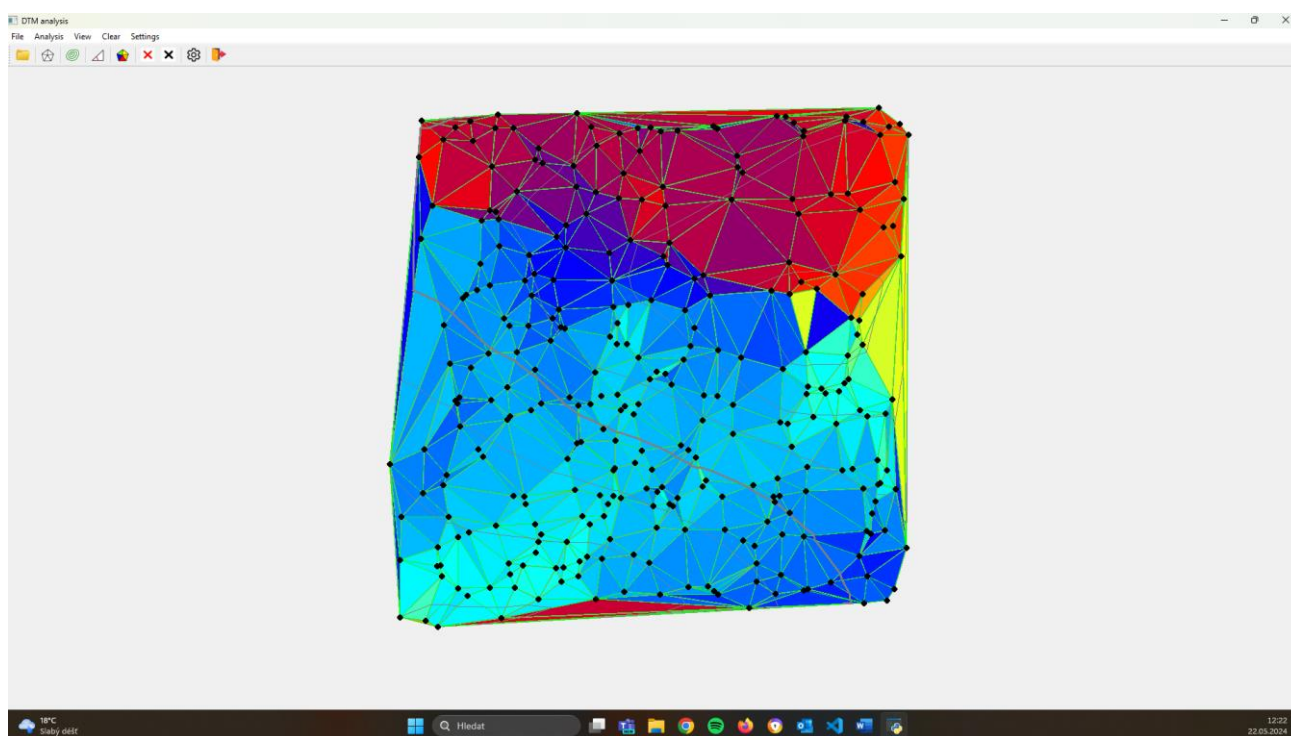


Obrázek 12: Sklon terénu

V obrázku lze pozorovat, že se jedná o poměrně rovný terén. Vrchol tedy není příliš prudce vystouplý nad okolním terénem a je možné, že by byl v terénu špatně rozpoznatelný. Lze ovšem pozorovat některá místa s prudším svahem. Například severní strana u vrcholu obsahuje lehce prudší přechod. Také lze vidět, že svah na jihozápad je výraznější než svah na přímý západ.

Stejně jako v předešlých částech lze pozorovat problematické části na okraji území. Například na východním okraji vzniká v tenkých trojúhelnících prudký svah, který zde ovšem určitě není a bylo by tedy vhodné okrajové trojúhelníky odstranit.

Poslední analýzou, kterou lze v aplikaci zobrazit, je orientace sklonu svahu. Tato orientace je zobrazena pomocí přechodu jednotlivých barev a legendu lze nalézt v kapitole Algoritmy. V následujícím obrázku lze pozorovat tuto analýzu vloženého území.



Obrázek 13: Orientace terénu

V obrázku lze pozorovat, že převážná část území je orientována jižně až jihozápadně. Na severní části obrázku je naopak orientace severní a v malé části na východě převažuje východní orientace. Z barevného zobrazení lze dobře pozorovat přechody mezi směry svahu.

Podobně jako v ostatních analýzách zde vzniká problém v okrajovém území, které by mělo být odstraněno. Oproti ostatním analýzám tu lze ovšem také pozorovat problém u vrcholu kopce. Zde se nepřírozně prohazuje modrá a žlutá barva. To je způsobeno rozložením bodů. Tyto body jsou kolem kopce rozloženy náhodně, a tudíž zde kopec nepřírozně klesá a stoupá. Pro správné rozložení by bylo vhodné mít jeden bod na vrcholu kopce a další body v okolí. V tomto zobrazení zřejmě chybí vrcholový bod.

Pro správnost výstupů lze data také porovnat se zobrazením vrstevnic v aplikaci Mapy.cz.



Obrázek 14: Testované území z Mapy.cz [2]

10 Přílohy

- Skript pro spuštění aplikace (*MainForm.py*)
- Skript pro vykreslování do aplikace (*draw.py*)
- Skript pro výpočet algoritmů (*algorithms.py*)
- Skript pro načtení dat (*pio.py*)
- Skript pro nastavení vykreslení vrstevnic (*Settings.py*)
- Skript pro třídu QPoint3DF (*QPoint3DF.py*)
- Skript pro třídu Edge (*Edge.py*)
- Skript pro třídu Triangle (*Triangle.py*)
- Složka s ikonami (*images*)
- Složka se vstupními daty (*mracno*)

11 Zdroje

- [1] https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk5_new.pdf
[2] <https://mapy.cz>

V Praze dne 22.5.2024

Jan Koudelka, Vojtěch Müller