

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

1	Zadání	2
1.1	Údaje o bonusových úlohách	3
2	Popis a rozbor problému	4
3	Popis použitých algoritmů	4
3.1	Delaunay triangulací	4
3.1.1	Vlastnosti	4
3.1.2	Implementace metody	4
3.2	Vrstevnice	5
3.2.1	Implementace metody	5
3.3	Sklon terénu	5
3.3.1	Implementace metody	5
3.4	Expozice terénu	5
3.4.1	Implementace metody	5
4	Informace o bonusových úlohách	6
5	Vstupní data	6
6	Výstupní data	6
7	Dokumentace	7
7.1	Třídy	7
7.1.1	Algorithms	7
7.1.2	Draw	8
7.1.3	SortByXAsc	9
7.1.4	Edge	9
7.1.5	QPoint3D	9
7.1.6	Triangle	9
7.1.7	Widget	10
8	Závěr	11
9	Reference	12

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 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 tři 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

Čas zpracování: 3 týdny

1.1 Údaje o bonusových úlohách

Nebyly vytvořeny žádné bonusové úlohy.

2 Popis a rozbor problému

Hlavním cílem této úlohy je tvorba aplikace, která nad výškopisnými body vytvoří Delaunay 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]

3 Popis použitých algoritmů

Existuje několik způsobů jak vytvořit triangulaci s různými kritérii. V této úloze jsme se zabývali Delaunay triangulací pomocí metody inkrementální konstrukce.

3.1 Delaunay triangulací

Delaunay triangulace je nejčastěji používanou triangulací při tvorbě digitálního modelu terénu. Delaunay triangulaci lze provádět v rovině i v prostoru.

3.1.1 Vlastnosti

Uvnitř opsané kružnice libovolného trojúhelníku triangulace neleží žádný jiný bod. Maximalizuje minimální úhel, avšak neminimalizuje maximální úhle v trojúhelníku. Vůči kritériu minimálního úhlu je lokálně i globálně optimální. Triangulace je jednoznačná pokud čtyři body neleží na kružnici.

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 ní 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 běží dokud struktura AEL není prázdná.

3.1.2 Implementace metody

1. Nalezení pivota q a k němu nejbližší bod: $q = \min(y_i)$
2. Hledání prvního Delaunay bodu.
3. Vytvoření prvního Delaunay trojúhelníku.
4. Hrany trojúhelníka se uloží do triangulace a do AEL.

5. Dokud není AEL prázdný proved'.
Hledej Delaunay bod k hraně z AEL.
Pokud Delaunay bod existuje.
Přidej nové hrany do DT.
Pokud nová hrana není v AEL přidej.

3.2 Vrstevnice

V úloze byly vrstevnice konstruovány lineární interpolací. U lineární interpolace je rozstup vrstevnice mezi dvěma body konstantní, tedy i spád. Při konstrukci vrstevnic hledáme průsečnici vodorovné roviny o výšce Z a rovinu trojúhelníka triangulace.

3.2.1 Implementace metody

1. Pro všechny hrany triangulace:
Otestuj zda hrana protíná vodorovnou rovinu o výšce Z :
Pokud protíná spočti polohové souřadnice.
Vytvoř hranu tvořící vrstevnici.

3.3 Sklon terénu

Skon terénu je definován jako úhle mezi normálovým vektorem $(0,0,1)$ a normálovým vektorem roviny trojúhelníku.

3.3.1 Implementace metody

1. Pro všechny trojúhelníky triangulace:
Vypočti sklon:

3.4 Expozice terénu

Expozice terénu je definována jako azimut k průmětu normálového vektoru roviny trojúhelníka do roviny x,y .

3.4.1 Implementace metody

1. Pro všechny trojúhelníky triangulace:
Vypočti expozici:

- 4 Informace o bonusových úlohách
- 5 Vstupní data
- 6 Výstupní data

7 Dokumentace

7.1 Třídy

7.1.1 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 Delaunay vlastnosti.

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

Metoda vytváří nad vektorem bodů Delaunay triangulaci, které se reprezentuje jako vektor hran.

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

Metoda vypočte průsečík hrany, tvořenou 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 $Z < z_{min}; z_{max} >$ s intervalem vrstevnic dz vrátí vektor hran definující vrstevnice.

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

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

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

Tato metoda slouží k vypočetní 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()
```

7.1.2 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 Delaunay triangulace z vykreslovacího okna.

```
void setDT(std::vector<Edge> &dt_)
```

Metoda slouží pro převod Delaunay triangulace do vykreslovacího okna.

void setContours(std::vector<Edge> &contours_)

Metoda slouží pro převod vrtevníc do vykreslovacího okna.

void setDTM(std::vector<Triangle> &dtm_)

7.1.3 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.

7.1.4 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.

7.1.5 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 výšce bodu.

void setZ(double z_)

Metoda nastaví výšku bodu.

7.1.6 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.

7.1.7 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.

8 Závěr

9 Reference

1. BAYER, Tomáš. Geometrické vyhledávání [online][cit. 1. 12.2018].
Dostupné z: <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk5.pdf>