

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## REKONSTRUKCE VÝŠKOVÉHO TERÉNU Z TURISTICKÉ MAPY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID TUŠKA

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **REKONSTRUKCE VÝŠKOVÉHO TERÉNU Z TURISTICKÉ MAPY**

LANDSCAPE RECONSTRUCTION FROM TOURIST MAP

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**DAVID TUŠKA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL ŠPANĚL**

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Tuška David**

Obor: Informační technologie

Téma: **Rekonstrukce výškového terénu z turistické mapy**

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte základy zpracování obrazu, zejména problematiku detekce hran a křivek v obraze.
2. Analyzujte současné přístupy k digitálnímu zpracování turistických map a jejich převod na výškovou mapu.
3. Vyberte vhodné metody a navrhnete jednoduchý nástroj pro převod turistických map na výškovou mapu.
4. Experimentujte s vlastní implementací a případně navrhnete vlastní modifikace metod.
5. Diskutujte dosažené výsledky a možnosti budoucího vývoje. Zvažte další pokračování v rámci diplomové práce.
6. Vytvořte stručný plakát prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucích.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Španěl Michal, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Pavel Zemčík  
vedoucí ústavu

**LICENČNÍ SMLOUVA  
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **David Tuška**  
Id studenta: 79282  
Bytem: Lamblova 113/25, 779 00 Olomouc  
Narozen: 10. 10. 1985, Olomouc  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Rekonstrukce výškového terénu z turistické mapy  
Vedoucí/školitel VŠKP: Španěl Michal, Ing.  
Ústav: Ústav počítačové grafiky a multimédií  
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracování díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnožení.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

  
.....  
Autor

## **Abstrakt**

Bakalářská práce se zabývá problematikou digitalizace turistických map. Cílem je získat z turistické mapy informace o vrstevnicích a výškových hodnotách v jednotlivých místech na mapě. Pomocí těchto informací se vytvoří výšková mapa, která je vhodná k dalšímu zpracování. Příkladem může být zobrazení trojrozměrného modelu krajiny, která může být použita při tvorbě počítačových her.

## **Klíčová slova**

rekonstrukce terénu, výšková mapa, počítačové vidění, zpracování obrazu, detekce hran, segmentace, turistická mapa

## **Abstract**

This work deals with problems of digitizing tourist map. The goal of this project is proposal and implementation computer program, which can be able to create heightmap from tourist map. Heightmap is a raster image used to store values of surface elevation. For example heightmaps are widely used in terrain rendering software and modern video games.

## **Keywords**

landscape reconstruction, heightmap, computer vision, image processing, edge detection, segmentation, tourist map

## **Citace**

David Tuška: Rekonstrukce výškového terénu z turistické mapy, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Rekonstrukce výškového terénu z turistické mapy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

David Tuška  
9. května 2008

## Poděkování

Rád bych poděkoval Ing. Michalu Španělovi za odborné vedení, účelné připomínky a konzultace.

© David Tuška, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Zpracování obrazu</b>	<b>3</b>
2.1	Co je to obraz . . . . .	3
2.2	Segmentace . . . . .	3
2.3	Detekce hran . . . . .	4
2.4	Gaussův filtr . . . . .	8
2.5	Výšková mapa . . . . .	9
<b>3</b>	<b>Návrh</b>	<b>10</b>
3.1	Nalezení vrstevnic v obraze . . . . .	10
3.2	Oprava získaných dat . . . . .	12
3.3	Značkování výšek vrstevnic . . . . .	15
3.4	Převod na výškovou mapu . . . . .	19
<b>4</b>	<b>Implementace</b>	<b>22</b>
4.1	Zobrazení výsledného 3D modelu . . . . .	22
4.2	Programová dokumentace . . . . .	22
<b>5</b>	<b>Zhodnocení dosažených výsledků</b>	<b>23</b>
5.1	Úspěšný převod na výškovou mapu . . . . .	23
5.2	Ukázky nedostatků vybraných metod . . . . .	23
<b>6</b>	<b>Závěr</b>	<b>28</b>



# Kapitola 1

## Úvod

Turistickou mapu držel v ruce téměř každý z nás. Avšak pouze někteří se v ní dokáží správně zorientovat. Nalezneme zde cesty, hrady, zámky, nejvyšší hory a nejhlubší jezera. Tato označení rozpozná správně většina výletníků. Představme si však následující situaci. Máme na výběr dvě možné cesty k našemu cíli. Jistě nás napadne otázka: „Která cesta je nejkratší?“ Musí nás však zajímat nejen vzdálenost, ale i schůdnost. Známý Murphyho zákon říká: „Zkratka je nejdelší vzdálenost mezi dvěma body“. Nejkratší cesta není vždy tou neoptimálnější. Najít nejlépe vyhovující trasu, která by nebyla přehnaně strmá a neschůdná, je mnohdy těžkým úkolem. Pokud nechceme následovat tisíckrát vyšlapané cesty, musíme se vyznat ve vrstevnicích a alespoň trochu v matematice.

Má bakalářská práce by měla tento problém ulehčit. Zabývá se návrhem a implementací algoritmů, které jsou schopné sestavit 3D model krajiny z turistické mapy reprezentované grafickým obrázkem. Trojrozměrné zobrazení krajiny je jistě uživatelsky zajímavější a následné hledání turisticky nejpříjemnější cesty může být pro uživatele dokonce zábavné.

Po prozkoumání současných přístupů k tématu mé bakalářské práce jsem objevil bakalářskou práci Daniela Bierzy: „Vytvoření výškové mapy z vrstevnic v rastrové mapě“.  
[1] Avšak po důkladném prozkoumání jsem došel k názoru, že je vhodné použití jiných metod pro zpracovávání vrstevnic z turistické mapy, chtěl bych se pokusit dospět k lepším výsledkům, než se doposud podařilo panu Birzovi. Hlavní rozdíl mého přístupu je nepřevádět vrstevnice na vektorové křivky, ale pracovat s nimi jako se skupinou bodů v rastrové grafice. Proces převodu na vektorové křivky je velmi náročný. Pokud po segmentaci 3.1 získáme neúplná data následně reprezentovaná přerušenými vrstevnicemi, vyjde snaha téměř naprázdno. Výsledek převodu nemá být přesný matematický model, ale výšková mapa.

První kapitola „Zpracování obrazu“ nás seznamuje se základními principy algoritmů, které se používají v počítačové grafice, především při práci s obrazovými daty. Druhá kapitola pojednává o návrhu programu pro převod turistické mapy na výškovou mapu, nalezneme zde analýzy metod, které byly použity k vytvoření výsledného programu. Ve třetí kapitole je zmíněna implementace programu, pojednává se zde o použité aplikaci pro výsledné zobrazení výškové mapy. Poslední kapitola se zabývá dosaženými výsledky a porovnáváním jednotlivých algoritmů.

## Kapitola 2

# Zpracování obrazu

Počítačová grafika je jednou z nejrychleji se rozvíjejících disciplín oboru informatiky. Můžeme ji rozdělit na dvě základní části. První je vytváření grafických obrazů, virtuálních scén a animací. Tato část je zastoupena v počítačových hrách a kreslících programech. Druhou oblastí, kterou se zabývá počítačová grafika, je zpracování obrazu a počítačové vidění. Výsledky je možné využít při rozpoznávání textu, řízení průmyslových robotů, porovnávání otisků prstů a rovněž pro rekonstrukci výškového terénu z turistické mapy.

### 2.1 Co je to obraz

„Matematickým modelem obrazu může být spojitá funkce  $f(i, j)$  dvou argumentů, souřadnic v rovině. Funkci  $f(i, j)$  se obvykle říká obrazová funkce. Hodnotou obrazové funkce je nejčastěji jas (intenzita). Jas je veličina, která souhrnně vyjadřuje vlastnosti obrazového signálu způsobem, který odpovídá jeho vnímání člověkem.“

„Obraz může být v jednodušším případě monochromatický. Je reprezentován jedinou obrazovou funkcí  $f(i, j)$ . Ve složitějším případě pracujeme s barevným (multispektrálním) obrazem. Každé dvojici plošných souřadnic  $(i, j)$  odpovídá vektor hodnot - např. jasů pro jednotlivé barevné složky obrazu.“ [2]

### 2.2 Segmentace

„Segmentace obrazu je jedním z nejdůležitějších kroků vedoucích k analýze obsahu zpracovávaných obrazových dat. Snahou je rozčlenit obraz do částí, které mají úzkou souvislost s předměty či oblastmi reálného světa zachyceného na obraze. Výsledkem má být soubor vzájemně se nepřekrývajících oblastí, které buď jednoznačně korespondují s objekty vstupního obrazu, pak jde o kompletní segmentaci, nebo vytvořené segmenty nemusí přímo souhlasit s objekty obrazu, a pak jde o částečnou segmentaci.“ [2]

Jedním z hlavních problémů, které ovlivňují segmentaci, je nejednoznačnost obrazových dat. Data jsou často doprovázena informačním šumem, kterého se snažíme pomocí segmentace zbavit a tím i výrazně redukovat objem zpracovávaných dat.

#### 2.2.1 Prahování

Jednou z nejjednodušších a nejstarších metod segmentace je prahování. Tato metoda je dnes v jednoduchých případech stále používána. Jedním z důvodů je její jednoduchost implementace a také rychlost.

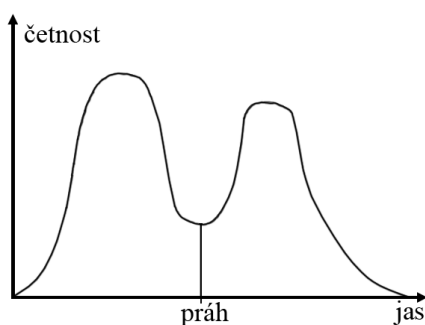
Cílem prahování je pro každý bod obrazu (pixelu) přiřadit hodnotu 1, pokud daný bod leží v hledané oblasti, nebo hodnotu 0 ostatním bodům (body pozadí). Prahování je založeno na předpokladu, že hledaná oblast má stejnou nebo velmi podobnou hodnotu jasu. Hledáme tedy body, jež mají jas v předem daném intervalu  $\langle a, b \rangle$  2.1. Druhou možností 2.2 implementace je zvolit si jakýsi práh  $t$ . Body, které mají hodnotu vyšší než zadaný práh, jsou označeny za součást hledané oblasti. Ostatní body jsou označeny jako body pozadí.

$$g(x, y) = \begin{cases} 1, & \text{pro } f(x, y) \in \langle a, b \rangle \\ 0, & \text{jinak} \end{cases} \quad (2.1)$$

nebo

$$g(x, y) = \begin{cases} 1, & \text{pro } f(x, y) \geq t \\ 0, & \text{pro } f(x, y) < t \end{cases} \quad (2.2)$$

„Úspěšnost prahování závisí na znalosti správné hodnoty prahu. Jestliže tuto hodnotu neznáme, je možné pokusit se ji stanovit na základě informací získaných z obrazu, který má být segmentován. Pro obrazy s bimodálním histogramem (histogram se dvěma vrcholy) jsou se např. často doporučuje volit jako hodnotu  $t$  prahu hodnotu, v níž histogram dosahuje mezi oběma vrcholy minima.“[3] (Obrázek 2.1)



Obrázek 2.1: Bimodální histogram jasu



Obrázek 2.2: Vlevo původní obrázek. Vpravo prahování s prahem 128. Převzato z [4]

## 2.3 Detekce hran

Hranu v obraze si můžeme představit jako hranici mezi dvěma oblastmi. Můžeme ji také najít na rozhraní světla a stínů. Hrana je vektorová veličina, která je určena velikostí



Obrázek 2.3: Prahování s optimálním prahem 74. Převzato z [4]

a směrem.

Formálněji se dá definovat hrana jako místo v obraze, kde dochází k velké změně jasové funkce. V tomto místě si můžeme povšimnout, že první derivace funkce jasu má vysokou hodnotu (Obrázek 2.4 Graf 2). Proto je nutné stanovit velikost prahu, který musí derivace v daném bodě přesáhnout, aby byl bod považován za hranu.

Z tohoto vyplývá: „Nejjednoduššími hranovými operátory jsou zjevně derivace  $\delta f/\delta x$  a  $\delta f/\delta y$ , které popisují změnu úrovně jasu ve směru os  $x$  a  $y$ . Těchto operátorů by bylo možné použít k hledání hran rovnoběžných se souřadnými osami. Při hledání hran obecného směru je zapotřebí vyšetřovat průběh jasu ve směru kolmém na směr potenciální hrany.“ [3] Kvůli jednoduššímu výpočtu se však hrany detekují pouze ve dvou, respektive ve čtyřech směrech.

Při praktické implementaci většinou nepoužíváme pro popis obrazů spojitě funkce, ale funkce diskrétní. Snažíme se proto derivace obrazové funkce aproximovat pomocí diferencí realizovaných diskrétní konvolucí. Operátory, které odhadují první derivaci, používají několik masek. Směr gradientu je možno odhadovat hledáním té masky, která odpovídá největší velikosti gradientu.

Druhou možností, jak hledat hrany v obraze, je hledat místo, kde druhá derivace obrazové funkce prochází nulou (Obrázek 2.4 Graf 3). Příkladem je Cannyho hranový detektor.

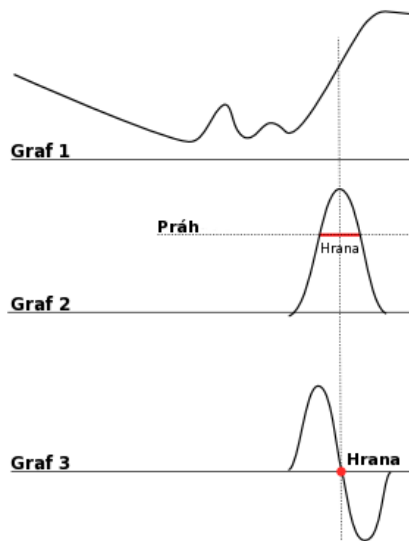
### 2.3.1 Robertsův operátor

Robertsův operátor je nejstarší a velmi jednoduchý hranový operátor, který využívá malé konvoluční masky (2x2), a tudíž je dosti citlivý na šum v obraze. Jeho konvoluční masky jsou:

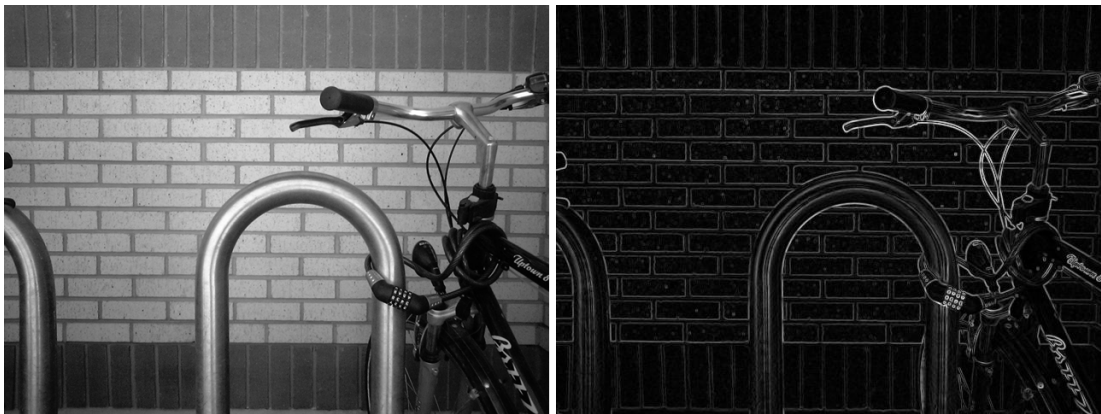
$$m_1 = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}, m_2 = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \quad (2.3)$$

Vlastní velikost hrany v bodě  $(x,y)$  se vypočítá podle funkce 2.4.

$$g(x,y) = \sqrt{(f(x,y) - f(x+1,y+1))^2 + (f(x+1,y) - f(x,y+1))^2} \quad (2.4)$$



Obrázek 2.4: Graf 1 Obrazová funkce - Hrana v obraze, Graf 2 První derivace obrazové funkce. Graf 3 - Druhá derivace. Převzato z [5]



Obrázek 2.5: Detekce hran pomocí Sobelova operátoru. Převzato z [10]

### 2.3.2 Sobelův operátor

Operátorů pro nalezení hran v obraze je celá řada - např. Prewittové, Robinsonův, Kirschův a Sobelův operátor. Jednotlivé operátory se liší především velikostí a koeficienty konvoluční masky. Velikost masky ovlivňuje především reakci na šum v obraze, čím větší konvoluční maska, tím je reakce na šum menší. Koeficienty konvoluční masky ovlivňují reakci na konkrétní vlastnosti obázku např. rychlost růstu gradientu nebo tvar hrany. „Bez znalostí statistických vlastností obrázku nelze předem říci, který z těchto operátorů bude lepší. Velmi často se vhodný operátor vybírá pokusem.“ [2]

Sobelův operátor je většinou definován pomocí dvou konvolučních matic (2.5) velikosti  $3 \times 3$ . Matice  $G_x$  je použita pro hledání hrany rovnoběžné s vodorovnou osou  $x$ , druhá matice  $G_y$  pro hledání hrany rovnoběžné s osou  $y$ . Pokud se na matice podíváme podrobněji, zjistíme, že první matice je pouze otočenou verzí druhé. Teoreticky bychom mohli matici  $G_x$  postupně otáčet o  $45^\circ$  a tím získat osm matic pro detekci hran v osmi směrech. Pro naše

účely nám ale postačí pouze dvě matice 2.5.

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.5)$$

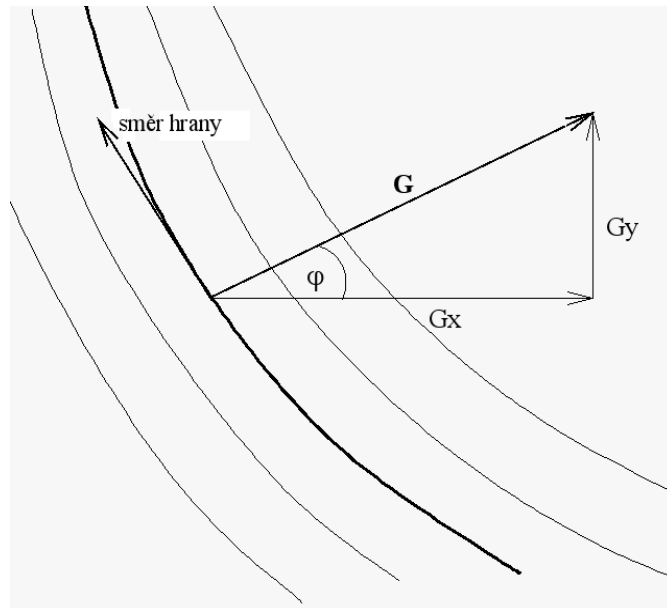
Pokud pomocí konvolučních matic 2.5 vypočteme velikost gradientu ve směru  $x$  a  $y$ , můžeme poté dle vzorce 2.6 vypočítat velikost gradientu  $G$  (Obrázek 2.6). Při implementaci hranového operátoru můžeme výpočet zjednodušit a počítat pouze přibližnou hodnotu gradientu jako součet absolutních hodnot gradientů ve směrech  $x$  a  $y$  2.7. Pro porovnávání velikosti gradientu s prahem, který určí zda je bod součástí hrany, nám přibližná přesnost postačí.

$$G = \sqrt{G_x + G_y} \quad (2.6)$$

$$G = |G_x| + |G_y| \quad (2.7)$$

Z obrázku 2.6 můžeme zjistit, že na základě  $G_x$  a  $G_y$  lze velmi jednoduše vypočítat směr gradientu a tím zároveň směrnici kolmice k samotné hraně.

$$\theta = \arctan \frac{G_x}{G_y} \quad (2.8)$$



Obrázek 2.6: Směr a velikost hrany

## 2.4 Gaussův filtr

Filtrace je jednou ze základních metod úpravy obrázků, může být využívána pro vyhlazení obrazu, rozostření obrazu, potlačení šumu, zvýraznění kontrastu a řady dalších úloh.

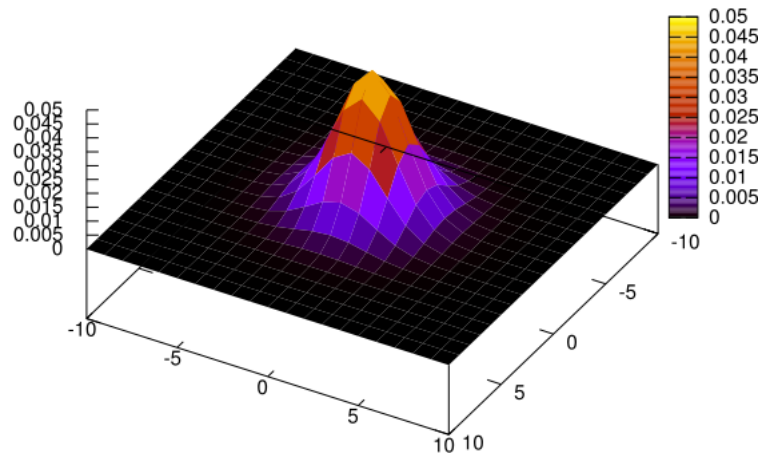
Gaussův filtr je nejpoužívanější filtr pro rozostřování obrázků a redukci šumu. Jeho funkce je založena na konvoluci obrázku s Gaussovou funkcí. Konvoluci provádíme pomocí konvoluční masky 2.10, která je složená z elementů určených Gaussovou funkcí.

Dvourozměrnou Gaussovou funkci definuje 2.9, kde  $\sigma$  je směrodatná odchylka Gaussovy funkce.

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{(u^2+v^2)}{2\sigma^2}} \quad (2.9)$$

Konvoluční maska (5x5) Gaussova filtru se směrodatnou odchylkou  $\sigma = 0,4$ :

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2.10)$$



Obrázek 2.7: Ukázka konvoluční matice Gaussovy funkce pro okolí 10 pixelů a směrodatnou odchylku  $\sigma = 1,8$ . Převzato z [9]

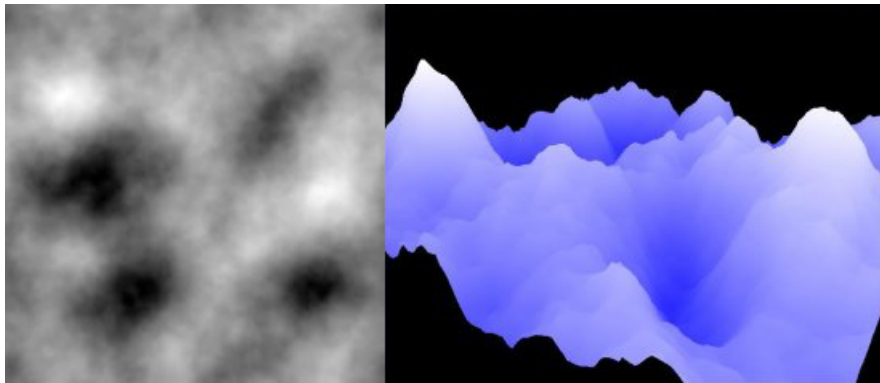


Obrázek 2.8: Ukázka použití Gaussova filtru s  $\sigma = 2$ . Převzato z [8]

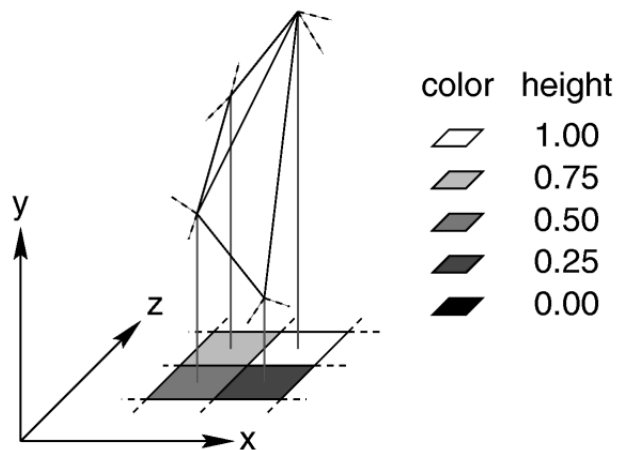
## 2.5 Výšková mapa

Výšková mapa (anglicky height fields nebo height map) je využívána v mnoha grafických aplikacích především pro reprezentaci terénu v hrách a simulačních programech, které zobrazují trojrozměrný model krajiny. Příklad využití výškové mapy při rekonstrukci terénu můžete vidět na obrázku 2.9.

Výšková mapa je grafická reprezentace funkce, která pro souřadnice  $x$  a  $y$  vrací souřadnici  $z$  dané mapy tj. informaci o výšce v daném bodě. Výšková mapa je zobrazena jako černobílý obrázek, kde černá barva reprezentuje nejnížší výšku (místo) a bílá barva reprezentuje nejvyšší výšku (místo) na mapě (Obrázek 2.10). Jiné zobrazení může být pomocí matice hodnot, kde každá buňka dané matice reprezentuje přesnou výšku terénu. [7]



Obrázek 2.9: Výšková mapa a následné zobrazení terénu. Převzato z [7]



Obrázek 2.10: Výšková mapa reprezentující výšku mapy. Převzato z [6]



# Kapitola 3

## Návrh

V následující kapitole se postupně seznámíme s jednotlivými kroky převodu rastrového obrázku na výškovou mapu.

Jednotlivé kroky:

- **Selekce vrstevnic** ze zdrojových dat pomocí filtrace vybraných barev.
- **Úprava nalezených vrstevnic** pro dosažení lepších výsledků a jednodušší práci s vrstevnicemi.
- **Přiřazení výšek vrstevnicím** po zadání několika vrcholů.
- **Převod na výslednou výškovou mapu**, která se může dále využít, a samotná rekonstrukci výškového terénu.

### 3.1 Nalezení vrstevnic v obraze

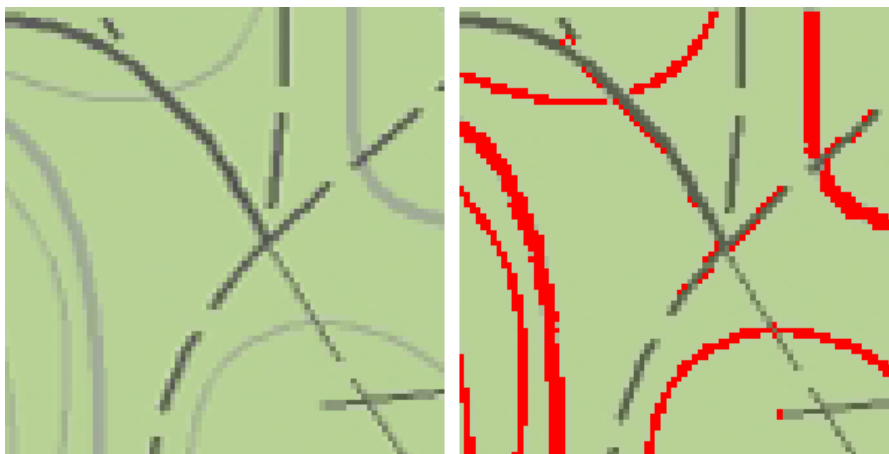
Vstupní rastrový obrázek obsahuje mnoho informací. Protože by se s takovým obrázkem pracovalo obtížně, musíme z něj nejprve získat jednotlivé vrstevnice. Ostatní informace zatím nebudeme využívat, pro tuto chvíli jsou pro nás nadbytečné.

#### 3.1.1 Filtrace podle barvy

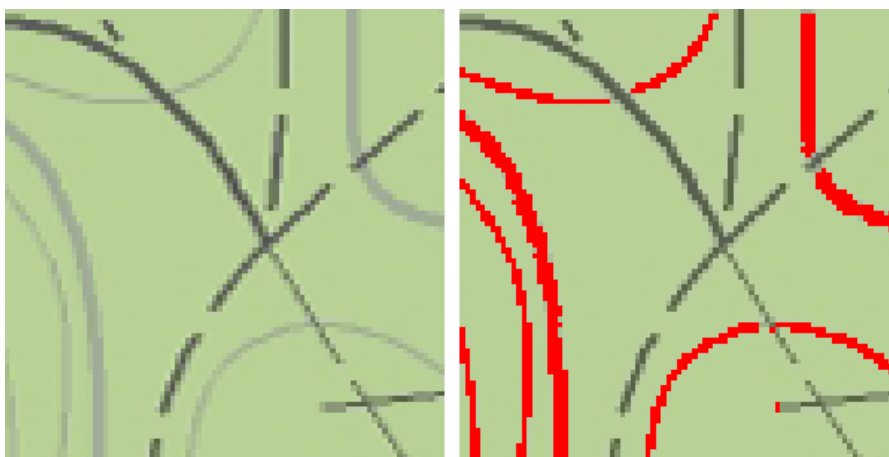
Nejjednodušší možný způsob získání vrstevnic je selekce dle vybrané barvy. Máme jednu nebo více zvolených barev a dovolenou odchylku od referenčních barev. Pro každý bod obrázku vypočteme podle vzorce 3.1 součet absolutních hodnot rozdílů jednotlivých barevných složek. Pokud vypočtená hodnota je menší než hodnota zvolené odchylky, prohlásíme daný bod za bod vrstevnice.

$$\text{odchylka} = |R_{ref} - R_{sel}| + |G_{ref} - G_{sel}| + |B_{ref} - B_{sel}| \quad (3.1)$$

Jelikož vrstevnice mohou mít místy odlišnou barvu, můžeme selekci podle barvy mírně upravit. Úprava spočívá v tom, že do zvolených barev zahrneme i barvy okolí bodu, jestliže jsou dané body okolí již součástí vrstevnice. Tato úprava nám sníží počet nutných zadávaných referenčních barev a tím i počet nutných zásahů uživatele. Stejného výsledku bychom mohli dosáhnout zvýšením dovolené odchylky, to by ale mohlo vést k tomu, že budou vybrány i nežádoucí body.



Obrázek 3.1: Selektce vybrala i okolí, které nepatří vrstevnici



Obrázek 3.2: Správná detekce vrstevnic

### 3.1.2 Nežádoucí okolí jiných objektů

V okolí jiných objektů, než jsou vrstevnice, můžeme najít stejné barvy, jež jsme vybrali pro selekci. Daný nežádoucí efekt můžeme vidět na obrázku 3.1. Proto upravíme filtraci podle barvy tak, že stanovíme jednu nebo více barev, které se nesmí vyskytovat v těsném okolí vrstevnice. Poté vždy zkoumáme, zda daný bod obrázku je v toleranci zvolené odchylky a pokud ano, zjistíme, zda-li v jeho blízkosti není jedna ze zakázaných barev okolí. Správnou selekci vrstevnic můžeme vidět na obrázku 3.2.

Přestože jsme eliminovali nežádoucí nadetekované body na minimum, nalezneme na obrázku osamocené body, které nepatří vrstevnicím. Proto objekty, jež jsou tvořeny méně než třemi pixely, nebudeme považovat za vrstevnice.

### 3.1.3 Detekce hran

Pro získání vrstevnic bychom mohli využít rovněž detekci hran. Metoda samotná by však odhalila i hrany, které nepatří jednotlivým vrstevnicím. K nalezení jednotlivých vrstevnic jsem proto použil selekci podle barvy. Jelikož jsem dosáhl pomocí této metody uspokojivých výsledků, nepokoušel jsem se již modifikovat selekci podle barvy s detekcí hran,

X	X	X	X	X
X	X	X	X	X
X				X
X				X

Obrázek 3.3: Matice pro nalezení koncových bodů

kde by detekce hran vytvářela mapu pravděpodobnosti. Následná selekce podle barvy by poté pracovala s mapou pravděpodobnosti tak, že by v místech, kde byla nalezena hrana, zvyšovala dovolenou odchylku od zvolené barvy vrstevnice.

### 3.1.4 Indexace

K dalšímu zpracování vrstevnic musíme jednotlivé nadetekované body seskupit do objektů. Docílíme toho tak, že budeme procházet jednotlivé body obrázku a pokud narazíme na bod, který je součástí vrstevnice a zároveň již není součástí nějakého objektu (nemá dosud přiřazeno žádné číslo), tak mu přiřadíme číslo o 1 vyšší než předchozímu objektu. Následně se podíváme na osmi-okolí kolem daného bodu. Jestliže je některý bod také neindexovaný bod vrstevnice, přiřadíme mu stejnou hodnotu jako předcházejícímu objektu. Daný princip opakujeme, dokud nenajdeme všechny body dané vrstevnice. Dále pokračujeme v procházení obrázku a hledání dalšího objektu.

## 3.2 Oprava získaných dat

Získat data pomocí segmentace se nikdy nepodaří bezchybně. Jednotlivé vrstevnice jsou často překryty jinými značkami a značeními. Tato část kapitoly by se proto měla zabývat opravou přerušovaných vrstevnic, aby se s nimi dalo následně pracovat co nejjednodušeji.

### 3.2.1 Nalezení koncových bodů

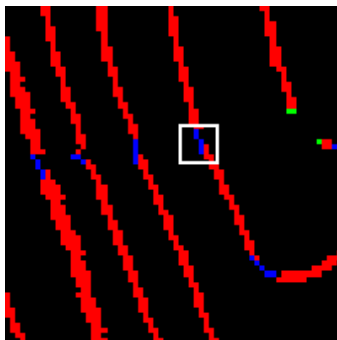
Spojování přerušovaných vrstevnic je založeno na principu spojování koncových bodů daných objektů. Nejprve tedy budeme muset najít koncové body jednotlivých vrstevnic.

Detekce koncových bodů je založena na postupném procházení obrázku po bodech a přiřádání matice (Obrázek 3.3) k jednotlivým bodům. Pokud je matice přiložena tak, že všechny body matice označeny pomocí  $X$  nezasahují do žádné vrstevnice, tak je prostřední bod matice označen jako koncový bod dané vrstevnice. Body matice označeny zelenou barvou nejsou pro posuzování, zda je daný bod koncovým bodem vrstevnice, důležité.

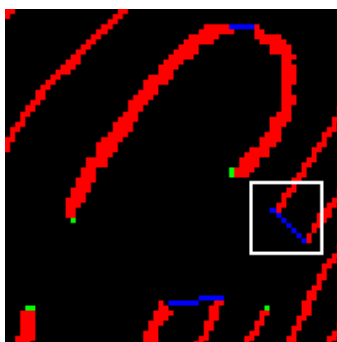
### 3.2.2 Spojování přerušovaných vrstevnic

Nyní jsme našli koncové body vrstevnic a můžeme se pokusit opravit přerušované vrstevnice. Nejjednodušším postupem, jak spojit poškozené části vrstevnic, je spojit nejbližší koncové body dvou různých vrstevnic. Postupným přiřádáním „plovoucího okna“ vybereme čtverec pixelů o rozměrech  $N \times N$ . s vybranými pixely následně pracujeme a rozhodneme, zda se mají poškozené části vrstevnic spojit.

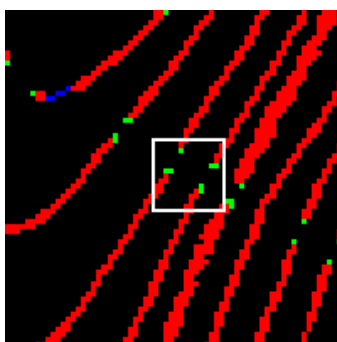
Pokud v rámci daného čtvercového výběru nalezneme koncové body, které patří dvěma vrstevnicím, můžeme dané body spojit. Tímto spojíme dvě poškozené vrstevnice v jednu. Podmínkou úspěšného spojení je, aby se v daném výběru vyskytovaly pouze dané dvě vrstevnice. Pokud by ve výběru byla ještě jiná vrstevnice, hrozilo by nesprávné spojení koncových bodů (Obrázek 3.6). Správné spojení koncových bodů můžeme vidět na obrázku 3.4.



Obrázek 3.4: Správné spojení vrstevnic (okno 7x7)



Obrázek 3.5: Chybné spojení vrstevnic (okno 15x15)



Obrázek 3.6: Nespojeno z důvodu možné chyby (okno 15x15)

Využitelnost jednoduché metody spojování nejbližších konců vrstevnic je omezená. V místech, kde je obrázek více porušen, je třeba použít větší rozměry „plovoucího okna“. To vede k tomu, že mohou být spojeny špatné koncové body jednotlivých vrstevnic. Nesprávné spojení můžeme vidět na obrázku 3.5.

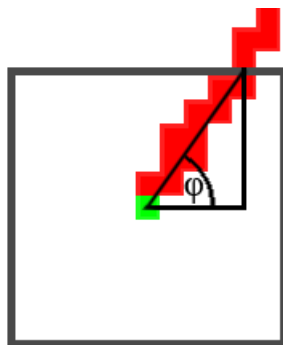
Lepších výsledků můžeme dosáhnout, pokud velikost okna budeme postupně zvětšovat. Avšak zvětšovat půjde pouze do předem dané velikosti okna, která je dle zkušeností je 12x12 pixelů.

Tuto metodu je vhodné využít především pro opravu drobných přerušení, které jsou způsobeny například řekami. Proto se podíváme na způsob spojování koncových bodů pomocí jejich směrnic.

### 3.2.3 Nalezení směrnice vrstevnice

Jak je vidět na obrázku 3.5, metoda spojování nejbližších koncových bodů má nevýhodu - mohou být spojeny koncové body, které s velkou pravděpodobností nepatří stejným vrstevnicím. Z tohoto důvodu by bylo výhodné zavést metodu spojování, která je založena na porovnávání směrnic vrstevnic. Podmínka spojení dvou vrstevnic by byla splněna, pokud právě dvě vrstevnice mají směrnice opačné.

Směrnici vypočítáme tak, že koncový bod umístíme do čtverce 5x5 pixelů (Obrázek 3.7). Zjistíme, ve kterých místech nám protíná daná vrstevnice čtverec. Z tohoto místa vedeme vektor do koncového bodu vrstevnice. Tento vektor je zároveň námi hledaná směrnice vrstevnice.



Obrázek 3.7: Výpočet směrnice v konci vrstevnice

Pokud je vrstevnice tlustší než 1 pixel, bude protínat čtverec na více místech vedle sebe. V takovém případě vytvoříme ze souřadnic daných bodů aritmetický průměr.

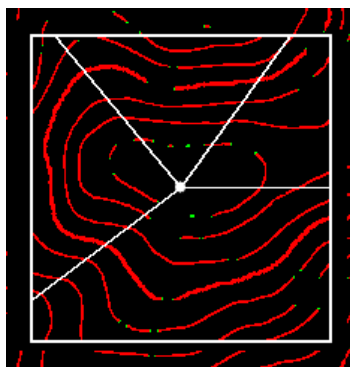
V místech, kde jsou vrstevnice velmi zakřiveny, může tato metoda určit směrnici nepřesně, což vede ke spojení křivek, které nepatří stejné vrstevnici. Pro správné napojení velmi zakřivených vrstevnic by bylo zapotřebí znát alespoň částečné informace o výškách jednotlivých vrstevnic. Proto se nejprve pokusíme přiřadit některým vrstevnicím výšku. K tématu spojování vrstevnic se vrátíme v podkapitole „Oprava vrstevnic na základě znalosti o výšce“ 3.3.2.

### 3.3 Značkování výšek vrstevnic

V kapitole 3.1 jsme získali ze vstupních dat informace o vrstevnicích. Snažili jsme se získaná data upravit 3.2, aby se s nimi lépe pracovalo. Nyní se pokusíme přiřadit vrstevnicím správnou výšku.

#### 3.3.1 Distribuce výšek vrcholů

Abychom mohli jednotlivým vrstevnicím přiřadit reálnou výšku, je třeba, aby uživatel zadal výšku několika vrcholů na mapě. Výšku ze zadaných vrcholů budeme distribuovat pomocí paprsků, které postupně vysíláme z vrcholů do okolí (Obrázek 3.8). Každému paprsku nejprve přiřadíme hodnotu odpovídající výšce daného vrcholu zaokrouhleného na desítky směrem dolů. Pokaždé, když se paprsek protne s vrstevnicí, tak přiřadíme dané vrstevnici aktuální hodnotu paprsku. Následně hodnotu paprsku snížíme o 10, což je výškový rozdíl jednotlivých vrstevnic na mapě. Tímto způsobem přiřadíme vrstevnicím, které leží nejbližší vrcholu, několik výškových hodnot. Z těchto hodnot můžeme poté vybrat tu nejčtenější nebo z daných hodnot vytvořit aritmetický průměr.



Obrázek 3.8: Kótování pomocí paprsků z vrcholu

Paprsky z jednotlivých vrcholů budeme vysílat jen do předem dané vzdálenosti od vrcholu. K základnímu okótování nejbližších vrstevnic nám bude stačit paprsky vysílat přibližně 70 až 150 pixelů od vrcholu. V ideálním případě bychom tedy vytvořili kružnici se středem ve vrcholu určité hory. Následně bychom vedli přímky z vrcholu k jednotlivým bodům kružnice. Dané přímky by nám poté reprezentovaly jednotlivé paprsky. Abychom nemuseli implementovat pohyb bodu po kružnici, spokojíme se s tím, že místo kružnice použijeme čtverec. Dosažený výsledek bude stejný a implementace pohybu bodu po čtverci (Obrázek 3.8), který budeme propojovat s vrcholem, bude mnohem jednodušší.

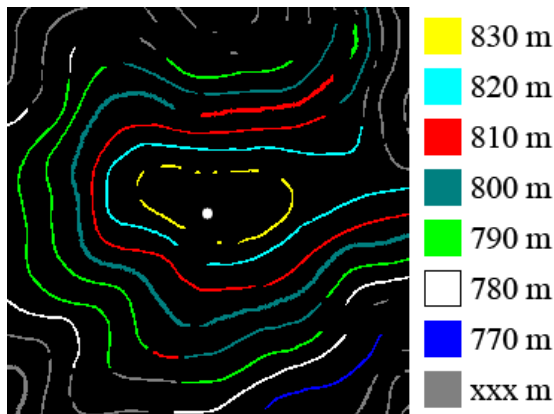
Tímto způsobem jsme získali základní okótování nejbližších vrstevnic okolo vrcholů.

#### 3.3.2 Oprava vrstevnic na základě znalosti o výšce

Nyní se pokusíme vrátit k opravě jednotlivých vrstevnic za pomocí nově získaných informací o výškách vrstevnic, které leží nejbližší vrcholům.

Od každého vrcholu si vybereme nejbližší vrstevnice tj. vrstevnice, které mají nejčtenější přiřazenou hodnotu rovnu výšce vrcholu zaokrouhlenou na desítky dolů. Například pro vrchol vysoký 837m vybereme všechny vrstevnice, které mají nejčtenější přiřazenou hodnotu

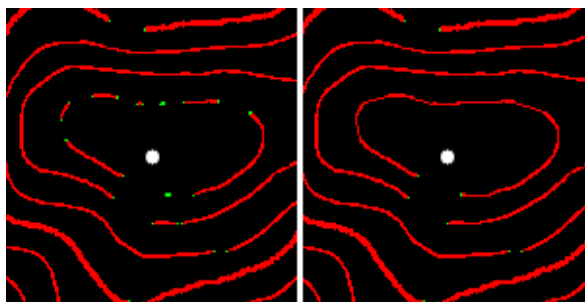
830m. Jelikož vybrané vrstevnice mají stejnou velikost, je velmi pravděpodobné, že to jsou části jedné přerušené vrstevnice. Proto se pokusíme dané části vrstevnice spojit.



Obrázek 3.9: Zvýraznění nejčtetnější přiřazené hodnoty po prvním okótování

Postupně se budeme pokoušet jednotlivé části vrstevnic spojit tak, že každé vrstevnici nalezneme nejbližší vrstevnici, se kterou danou vrstevnici spojíme. Pro každé dvě vrstevnice si tedy vypočítáme jejich vzájemnou vzdálenost. Využijeme informaci o koncových bodech, vzdálenosti budeme počítat pouze mezi těmito body. Jednotlivé vzdálenosti mezi vrstevnicemi seřadíme vzestupně, takto vybereme nejbližší dvě vrstevnice, které spojíme.

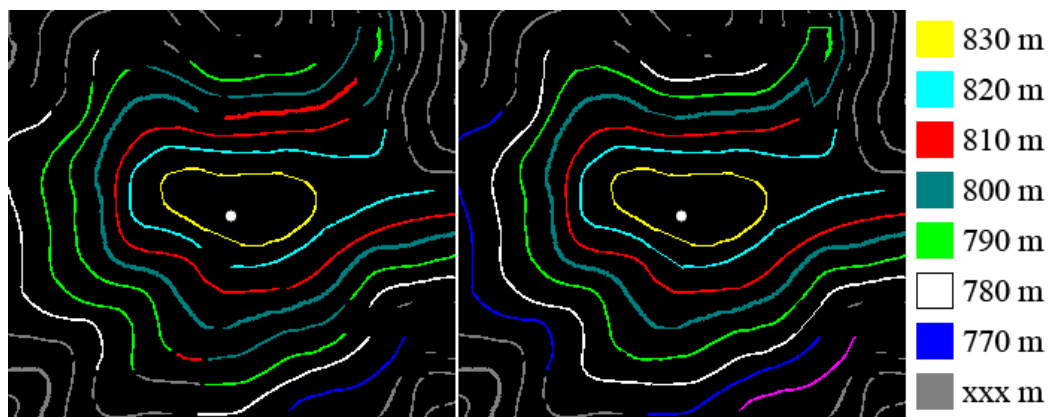
Pro spojení dvou částí vrstevnic máme dvě základní podmínky. První podmínkou je stejná přiřazená výška spojovaných částí, která je zajištěna výběrem vrstevnic, nad nimiž provádíme dané operace. Další nutnou podmínkou je, že při propojování jednotlivých vrstevnic není překřížena žádná jiná vrstevnice. Po propojení dvou křivek v jednu vrstevnici se pokusíme znovu najít koncové body vrstevnice tak, aby se odrušily koncové body v okolí spojení. Algoritmus spojování jednotlivých částí vrstevnic opakujeme tak dlouho, dokud nám nezůstane pouze jedna vrstevnice, nebo nejde vytvořit propojení, které by neporušilo základní pravidla propojení.



Obrázek 3.10: Oprava nejbližší vrstevnice

Pokud nám po spojování zůstane pouze jedna vrstevnice, můžeme se domnívat, že je elipsovitého tvaru (Obrázek 3.10). Můžeme se pokusit danou elipsu uzavřít. Abychom zjistili, zda je vhodné koncové body vrstevnice spojit, lze využít směrnice vrstevnice v koncových bodech 3.2.3. Pokud jsou jednotlivé směrnice v koncových bodech opačné, nebo alespoň s přesností  $30^\circ$ , můžeme koncové body dané vrstevnice spojit a takto pádem uzavřít vrstevnici.

Takto jsme opravili nejbliže ležící vrstevnici každého vrcholu. Nyní se pokusíme daný postup opakovat a opravit i další vrstevnice okolo vrcholu. Aby se nám nekumulovala chyba z předchozích kótování, odstraníme všem vrstevnicím přiřazené hodnoty. Docílíme toho, že následující kótování bude přesnější. Celkový postup budeme opakovat přibližně 5 až 10 krát. Výsledek opravy vrstevnic po opakování algoritmu je vidět na obrázku 3.11.

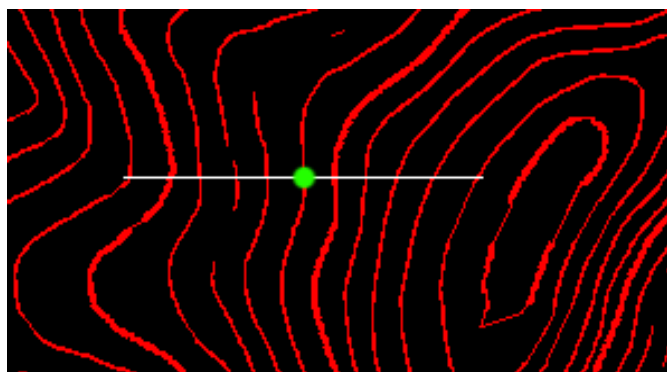


Obrázek 3.11: Oprava vrstevnic po 1. a 6. iteraci algoritmu založeného na znalosti výšky vrstevnic

### 3.3.3 Distribuce výšek mezi vrstevnicemi

Jelikož přesnost kótování pomocí paprsků z vrcholu 3.3.1 s rostoucí vzdáleností od vrcholu klesá, je nutné, abychom se zaměřili i na jiné metody kótování.

Pomocí metody „vysílaných paprsků“ z vrcholu 3.3.1 se nám podařilo správně okótovat vrstevnice, které leží nejbliže vrcholu. Protože potřebujeme okótovat i vzdálenější vrstevnice, bylo by vhodné distribuovat výšku i mezi vrstevnicemi dále od vrcholu. Z každého bodu vrstevnice povedeme kolmici na její směr vrstevnice (Obrázek 3.12 ) a budeme se snažit distribuovat výšku oběma směry.



Obrázek 3.12: Distribuce výšky mezi vrstevnicemi



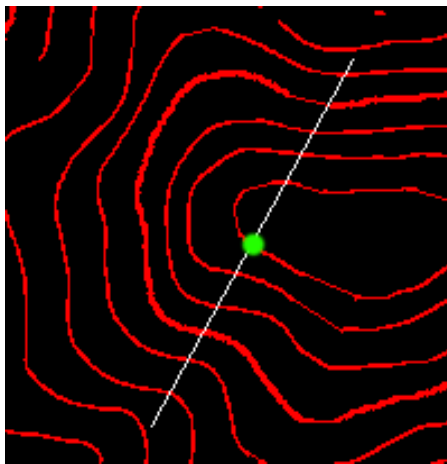
Na základě přiřazených hodnot zjistíme, že některé vrstevnice jsou okótovány lépe, jiné hůře. Pravděpodobnost „správného“ okótování vypočítáme jako poměr nejčtetnější přiřazené hodnoty ku celkovému počtu přiřazených hodnot. Například můžeme nalézt vrstevnici, která bude mít nejčtetnější výšku 680m zastoupenou 75×, celkově bude přiřazeno vrstevnici 100 hodnot. Pravděpodobnost, že vrstevnice má výšku 680m, je 75%.

Začneme postupně zpracovávat vrstevnice podle pravděpodobnosti okótování. Nejprve ty, které mají pravděpodobnost nejvyšší, poté vrstevnice s nižší pravděpodobností. Pravděpodobnost zpracovaných vrstevnic by neměla klesnout pod 75%. Pokud bychom zpracovávali vrstevnice s pravděpodobností nižší než 75%, zvyšovala by se pravděpodobnost, že vrstevnice je nesprávně okótovaná, což by mělo za následek distribuci chyby na okolní vrstevnice.

Pro výpočet kolmic k dané vrstevnici použijeme Sobelův operátor 2.3.2, který se využívá pro detekci hran. Lze jej využít také k výpočtu kolmice na hranu v daném bodě 2.8. Pro výpočet Sobelova operátoru se použije původní obrázek turistické mapy, pokud bychom výpočet prováděli z obrázku, ve kterém jsou zvýrazněny pouze vrstevnice, byl by výpočet méně přesný. Abychom nemuseli provádět výpočet Sobelova operátoru pro celý obrázek, budeme provádět výpočet pouze v nejbližším okolí dané vrstevnice v místě kde je předpokládána hrana obrazu.

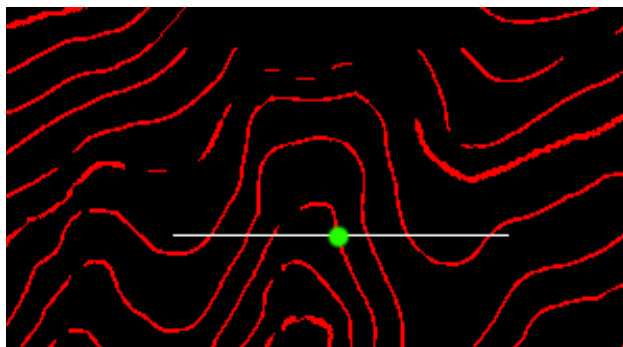
Kolmici z hrany vrstevnice vedeme obousměrně do vzdálenosti přibližně 75 pixelů. Vrstevnicím, které kolmicí protneme, se budeme snažit přiřadit „správnou“ výšku odvozenou z vrstevnice, ze které vedeme kolmicí.

Při distribuci a rozpočítávání výšky na sousední vrstevnice může nastat několik případů. Nejjednodušší případ rozpočítávání vidíme na obrázku 3.12, kde výšky všech protnutých vrstevnic tvoří ryze monotónní posloupnost (tj. klesající nebo rostoucí posloupnost). Snažíme se tedy najít směr posloupnosti a následně v daném směru rozpočítat výšky sousedních vrstevnic.



Obrázek 3.13: Distribuce výšky mezi vrstevnicemi - přechod přes stejnou vrstevnici

Složitější případ kótování můžeme spatřit na obrázku 3.13. Při kótování dojde k tomu, že jedna vrstevnice je protnuta dvakrát, v tomto případě vrstevnice netvoří monotónní posloupnost. V místě, kde je stejná vrstevnice protnuta dvakrát za sebou se bude měnit směr posloupnosti. Bude zde lokální minimum nebo maximum posloupnosti.



Obrázek 3.14: Distribuce výšky mezi vrstevnicemi - špatné okótování kvůli přerušení

Problém nastane, pokud se nedá místo, kde je vrstevnice protnuta dvakrát, přesně lokalizovat z důvodu přerušení vrstevnic (Obrázek 3.14). Dochází k chybnému okótování, které může zhoršit výsledný převod na výškovou mapu.

Po zpracování každé vrstevnice přepočítáme pravděpodobnosti okótování jednotlivých vrstevnic. Označíme si vrstevnici, kterou jsme již zpracovali, a pokračujeme dokud nezpracujeme všechny ostatní nebo dokud nenalezneme vrstevnici, která by měla pravděpodobnost okótování větší než 75%. Tímto postupem by se nám mělo podařit okótovat i vrstevnice, které leží dále od vrcholů.

### 3.4 Převod na výškovou mapu

V tento okamžik máme okótovanou většinu vrstevnic a můžeme je začít převádět na cílovou výškovou mapu. Nejprve ale musíme všem obrázkovým bodům přiřadit výšku.

#### 3.4.1 Výpočet výšky pomocí paprsků od vrcholu

Prozatím jsme přiřazovali výškovou hodnotu pouze vrstevnicím. Nyní začneme přiřazovat výšku také jednotlivým bodům vstupního obrázku. Převod na výškovou mapu začneme podobně jako při kótování vrstevnic z vrcholů 3.3.1. Tentokrát budeme vysílat paprsky do větší vzdálenosti než při kótování, protože již máme vrstevnice opraveny a okótovány.

Pokud vyšleme paprsek z vrcholu, protne nám několik vrstevnic. Tím se nám paprsek rozdělí na několik částí. Jednotlivé části budeme zpracovávat postupně. Začneme prvním úsekem, který vede od samotného vrcholu k první protnuté vrstevnici. Zjistíme délku daného úseku  $d$  (počet pixelů na úseku) a výškový rozdíl  $\Delta h$ , který vypočítáme jako rozdíl vrcholu a dané vrstevnice. Pomocí vzorce 3.2 vypočteme strmost daného úseku. Ta nám udává výškový rozdíl mezi dvěma sousedními pixely. Jednotlivým bodům pomocí strmosti přiřadíme vypočtenou výšku. Tento postup opakujeme na dalších úsecích dále od vrcholu.

$$strmost = \frac{\Delta h}{d} \quad (3.2)$$

#### 3.4.2 Výpočet výšky mezi vrstevnicemi

Pomocí předchozí metody se nám podařilo převést vrcholy hor a jejich blízké okolí na výškovou mapu. Abychom převedli i zbytek mapy, použijeme metodu obdobnou metodě kolmic, jež jsme použili při kótování 3.3.3.

Metoda paprsků nám k jednotlivým bodům přiřadila 0 až  $n$  výškových hodnot. Postupně projdeme všechny vrstevnice, pokud bude mít některý bod vrstevnice přiřazeno méně jak pět hodnot, tak z daného bodu povedeme kolmici k vrstevnici. Kolmice nám protnou vrstevnice a pro získané úseky rozpočítáme výškové hodnoty podobně jako u předchozí metody.

Nakonec vypočítáme pro každý obrázkový bod průměr přiřazených výšek. Jednotlivé hodnoty bychom mohli uložit jako matici do textového souboru, tím získáme jeden možný způsob reprezentace výškové mapy.

### 3.4.3 Uložení výškové mapy do obrázku

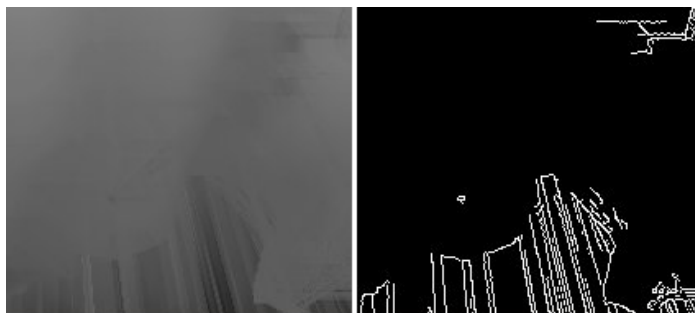
Nejpoužívanější formát výškové mapy (popsaný v kapitole 2.5) je černobílý obrázek, který obsahuje 256 stupňů šedi. Pro každý bod obrázku převedeme vypočítanou výšku na intenzitu šedi pomocí vzorce 3.3, kde  $h_{min}$  a  $h_{max}$  jsou minimální a maximální výšky na dané mapě.

$$int = \frac{h - h_{min}}{h_{max} - h_{min}} * 255 \quad (3.3)$$

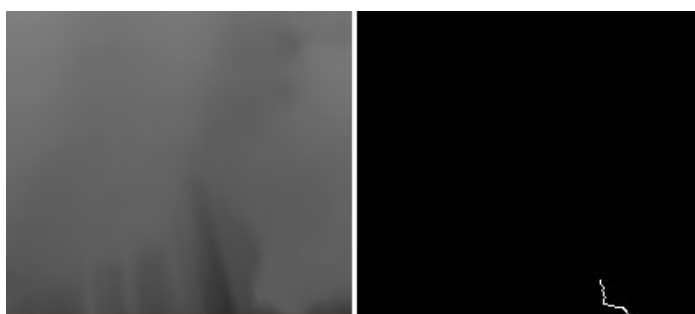
### 3.4.4 Vyhazení výškové mapy

Výsledná výšková mapa by měla splňovat základní dvě podmínky. První podmínkou je přesnost převodu zadané mapy na cílovou výškovou mapu, druhou, že výšková mapa by měla tvořit hladký gradient bez hran.

Na obrázku 3.15 vidíme ukázkou okrajové části mapy, kde při převodu na výškovou mapu došlo k drobné chybě, která byla způsobena tím, že na okraji mapy již nemáme dostatečné informace pro výpočet správné výšky. Aby byla výsledná výšková mapa lépe použitelná zbavíme se prudkých přechodů pomocí Gaussova filtru 2.4. Filtr budeme aplikovat na celý výsledný obrázek. Místa, která nejsou poškozená, by Gaussov filtr měl ovlivnit minimálně. Výsledné použití Gaussova filtru pro vyhlazení můžeme vidět na obrázku 3.16.



Obrázek 3.15: Vlevo část poškozené výškové mapy. Vpravo detekce hran pro určení chybného převodu na výškovou mapu



Obrázek 3.16: Použití Gaussova filtru pro vyhlazení výsledné výškové mapy

# Kapitola 4

## Implementace

Pro implementaci aplikace jsem vybral programovací jazyk C++ především z důvodu přenositelnosti mezi OS MS Windows a Linux. Pro zpracovávání obrazových dat a snadnější práci s nimi jsem použil grafickou knihovnu OpenCV (Open Source Computer Vision Library) vyvíjenou firmou Intel a dostupnou na internetové stránce <http://www.intel.com/technology/computing/opencv/>. Hlavním důvodem mého rozhodnutí bylo přívětivé rozhraní knihovných funkcí a jednoduché vestavěné GUI (Grafické uživatelské rozhraní).

Při návrhu aplikace je kladen důraz na oddělení logické části programu od uživatelského rozhraní. To nám dovoluje jednoduše přepracovat uživatelské rozhraní bez nutného zásahu do samotné aplikace. Detailnější informace naleznete v programové dokumentaci [4.2](#).

### 4.1 Zobrazení výsledného 3D modelu

Pro výsledné zobrazení výškového modelu byla použita aplikace vytvořená na základě internetového tutoriálu Nehe (<http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=45>). Aplikace je založena na OpenGL a využívá rozšíření moderních grafických karet nazývané „Vertex Buffer Object“. Pokud grafická karta nepodporuje dané rozšíření, je pro vykreslení použita pomalejší metoda pomocí „Vertex Arrays“.

Převzatá aplikace zpracovává výškovou mapu ve formě černobílého obrázku uloženého ve formátu BMP (Microsoft Windows Bitmap). Na přiloženém CD můžete najít upravenou aplikaci, která kromě výškové mapy dokáže načíst také obrázek s texturou, která se model krajiny namapuje.

### 4.2 Programová dokumentace

Programovou dokumentaci lze nalézt na přiloženém CD v adresáři prog\_doc.

## Kapitola 5

# Zhodnocení dosažených výsledků

Tato kapitola se zabývá zhodnocením výsledků bakalářské bakalářské práce. Nejprve se podíváme na úspěšný pokus převodu turistické mapy na výškovou mapu, následně se zaměříme na nedostatky vybraných použitých metod.

### 5.1 Úspěšný převod na výškovou mapu

Na obrázcích 5.1 až 5.5 si můžeme všimnout relativně dobrých výsledků. Selektce 5.2 s následným nalezením koncových bodů se podařila velmi dobře.

Na obrázku 5.3 je znázorněno okótování vrstevnic. Každé výškové hodnotě je přiřazena barva podle dané výšky (desítek hodnoty výšky). Vrstevnice těsně kolem vrcholu jsou správně opraveny a okótovány. Vzdálenější vrstevnice mohou být hůře okótovány z důvodu přerušení vrstevnic. Bylo by vhodné pomocí znalosti o výšce opravovat vrstevnice nejen kolem vrcholu, ale i ty vzdálenější. Opravování vzdálenějších vrstevnic nese mimo jiné i riziko, že dojde k chybnému spojení distribuujícího se i dále.

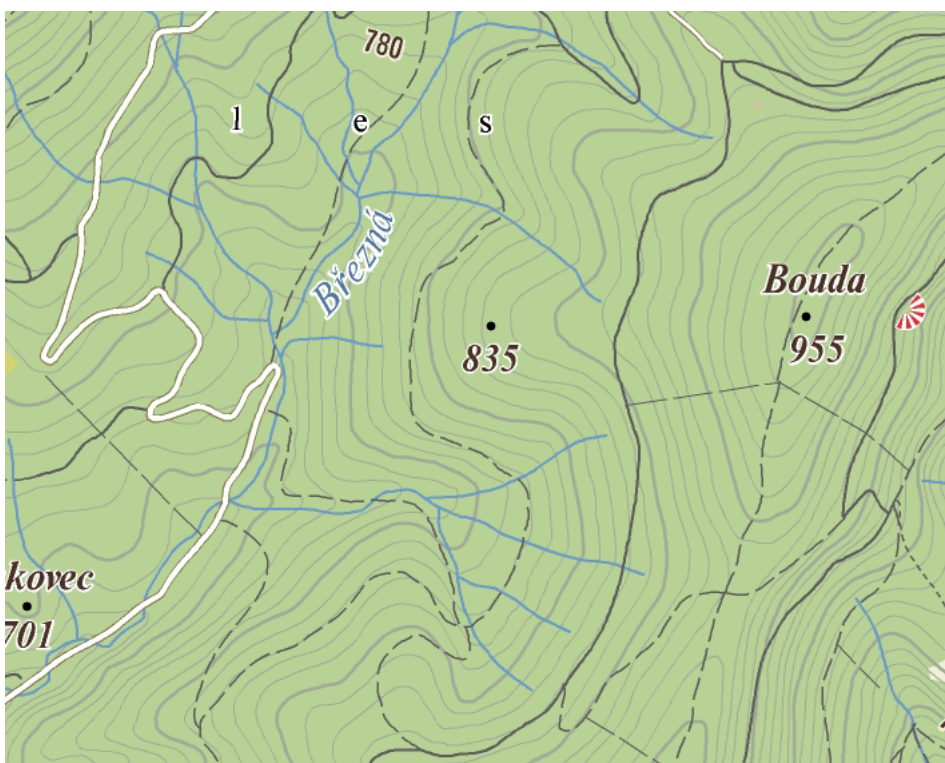
Výška jednotlivých vrstevnic je lineárně rozpočítána do okolních bodů a převedena na výškovou mapu, výsledek převodu můžeme vidět na obrázku 5.4.

Na základě původní turistické mapy a vytvořené výškové mapy můžeme vytvořit trojrozměrný model krajiny (Obrázek 5.5).

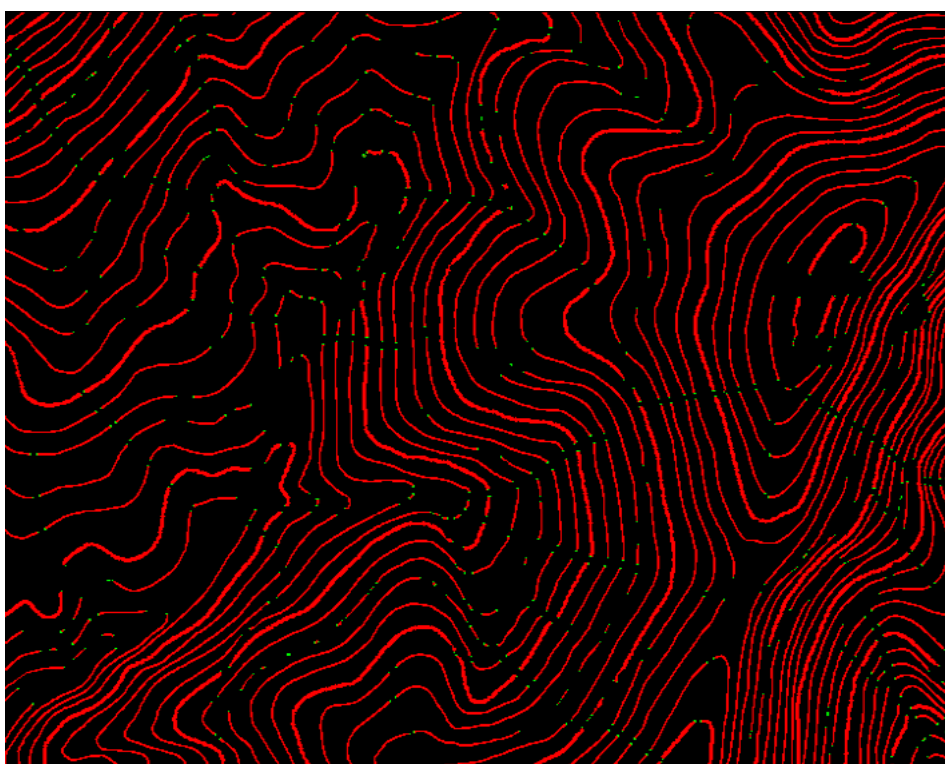
### 5.2 Ukázky nedostatků vybraných metod

Pokud na mapě leží vrstevnice blízko u sebe, může metoda selektce vrstevnic selhat a spojit několik vrstevnic v jednu 5.7. Při kótování vrstevnic s daným problémem nepočítáme, proto vede ke špatným výsledkům převodu výškové mapy.

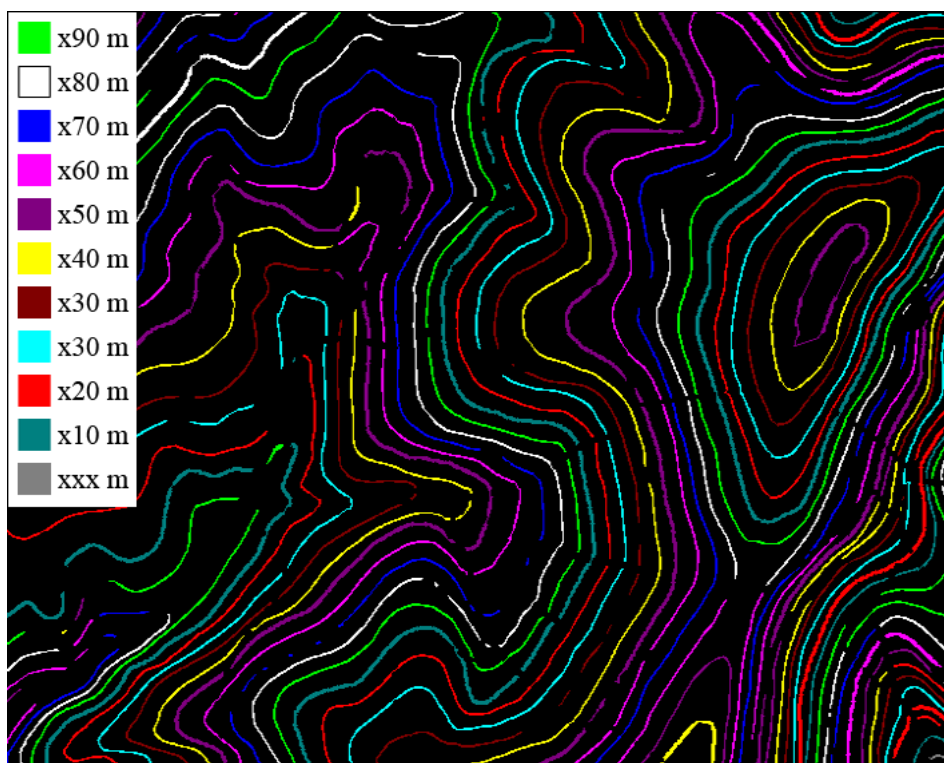
Na obrázku 5.8 vidíme, že v některých případech oprava vrstevnic pomocí výšky selhává. Toto selhání má za následek velmi špatné výsledky kótování i následného převodu na výškovou mapu. Bylo by proto vhodné opravit tuto metodu. Například využít informaci o směrnících spojovaných částí vrstevnic. Směrníc jde však využít jen při drobném porušení vrstevnic. Pokud jsou vrstevnice přerušeny více, jednotlivé konce vrstevnic na sebe nemusí směřovat. Z tohoto důvodu jsem metodu prozatím neupravoval.



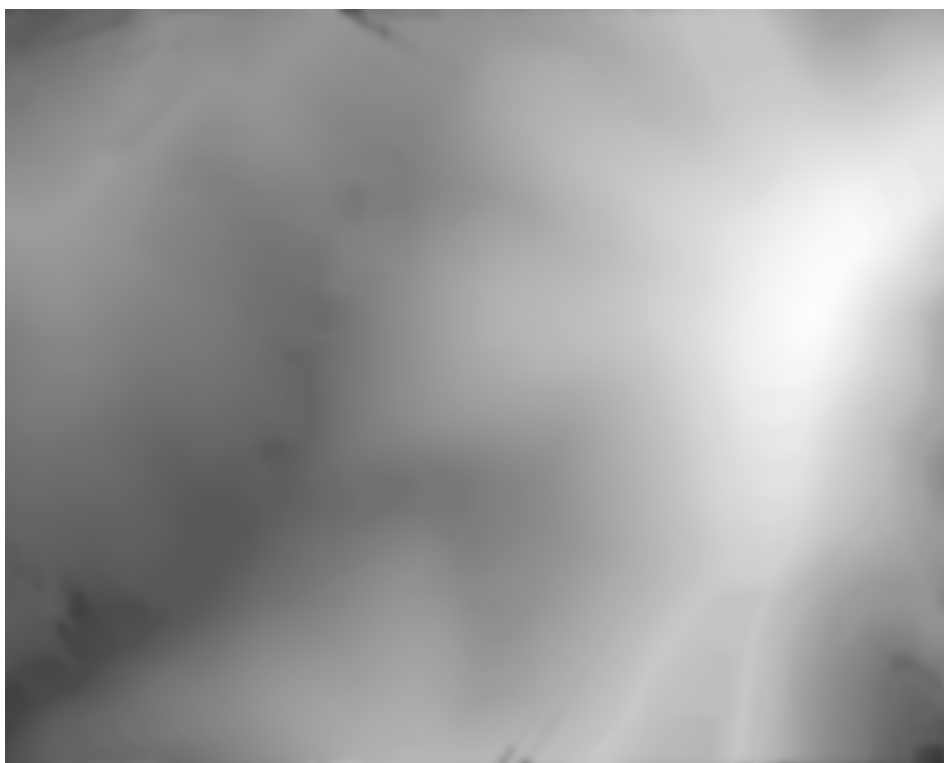
Obrázek 5.1: Vstupní turistická mapa



Obrázek 5.2: Selektce vrstevnic a detekce koncových bodů (koncové body zobrazeny zeleně)

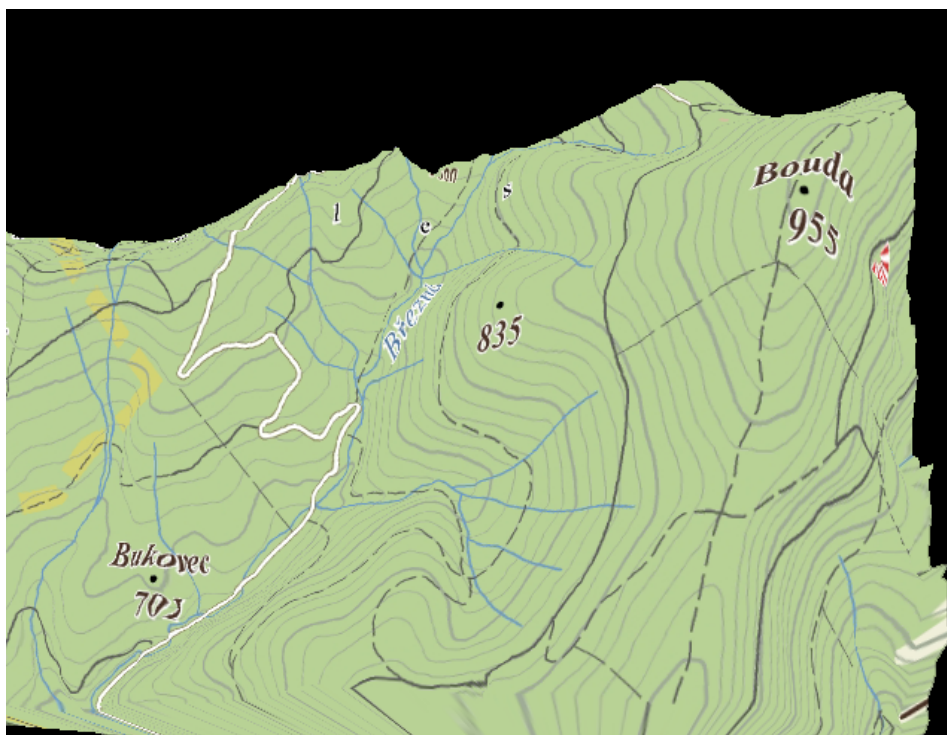


Obrázek 5.3: Okótování vrstevnic po zadání výšky jednoho vrcholu

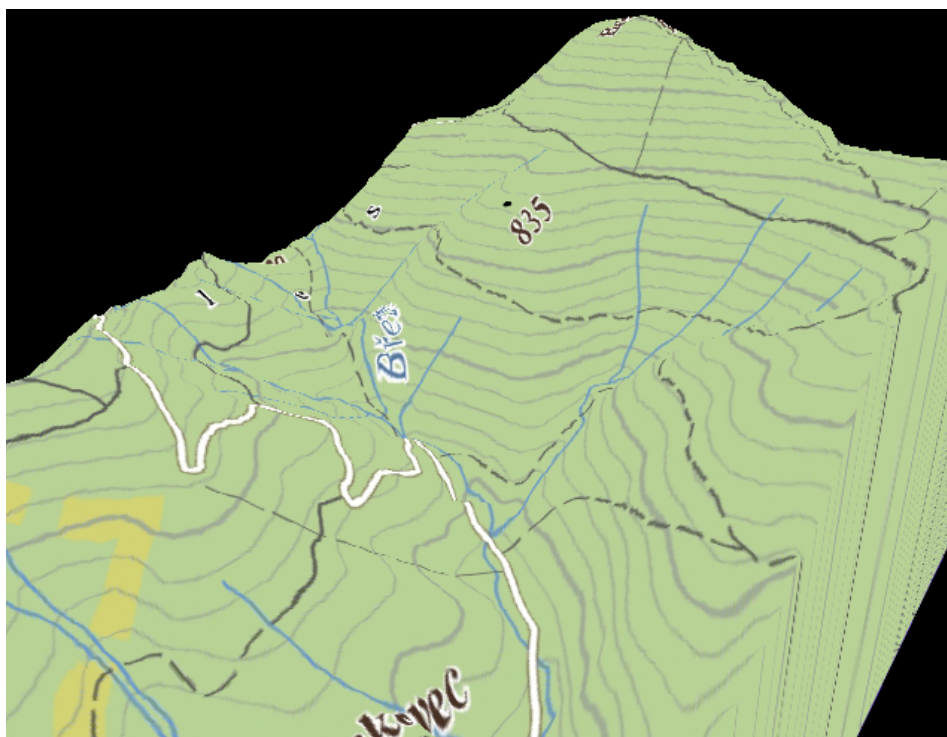


Obrázek 5.4: Převod na cílovou výškovou mapu

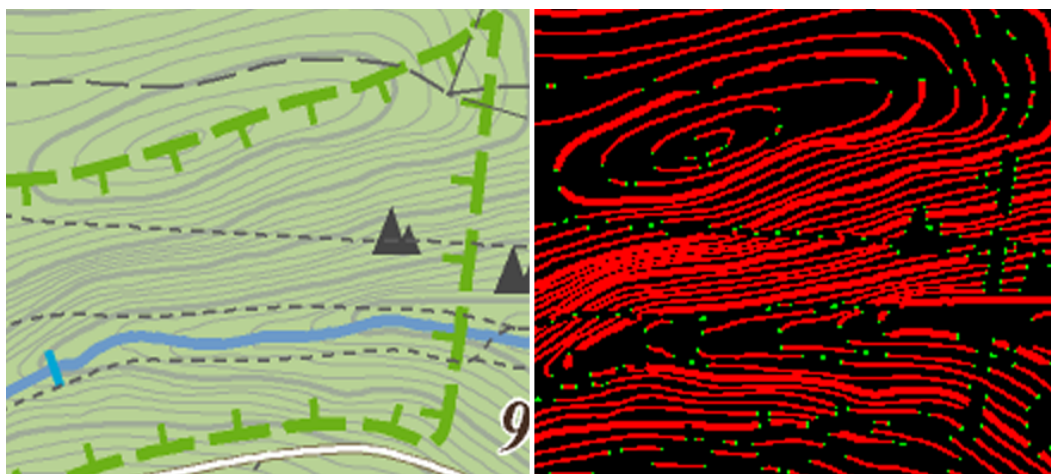




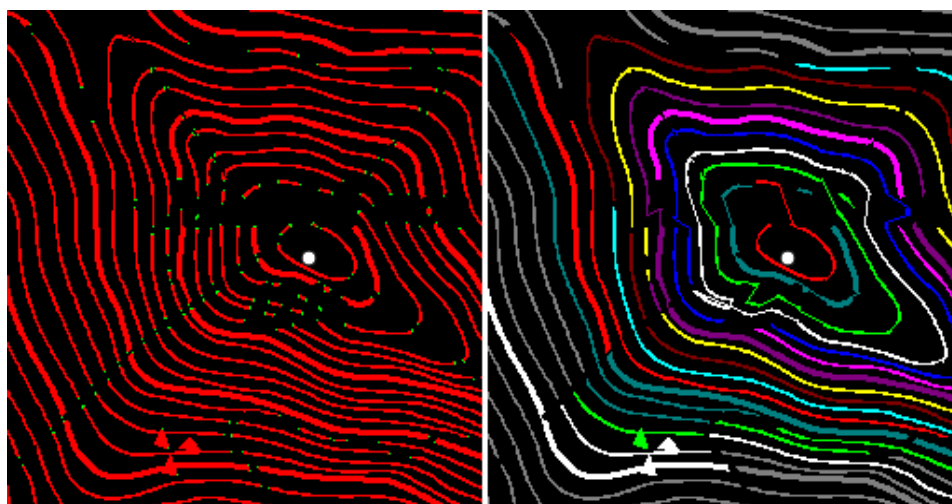
Obrázek 5.5: Zobrazení 3D modelu krajiny z výškové mapy I



Obrázek 5.6: Zobrazení 3D modelu krajiny z výškové mapy II



Obrázek 5.7: Selhání selekce vrstevnic - několik vrstevnic spojených v jednu



Obrázek 5.8: Špatné spojení vrstevnic na základě znalosti o výšce

# Kapitola 6

## Závěr

Cílem mé bakalářské práce bylo vytvořit aplikaci, která na základě rastrového obrázku turistické mapy vytvoří výškovou mapu terénu. Požadovaného cíle se mi podařilo dosáhnout. Pro „jednoduchou“ turistickou mapu aplikace dokáže vytvořit odpovídající výškovou mapu.

Převod na výškovou mapu se provádí pomocí posloupnosti jednotlivých kroků - selekce, oprava vrstevnic, kótování, převod na výškovou mapu.

Selekci jsem provedl pomocí upravené metody „filtrace podle barvy“ vrstevnice 3.1. I přes jednoduchost dosahovala metoda velmi dobrých výsledků. Oprava vrstevnic je základním krokem ke správnému okótování. Pro místa, kde jsou vrstevnice přerušeny jen málo, je vhodné použít metodu, která hledá nejbližší koncový bod jiné vrstevnice 3.2.2. Oprava vrstevnic při použití znalosti o výšce 3.3.2 je účinná v blízkosti vrcholů, kde nedochází k velké distribuci chyb. Tato metoda nedosahuje vždy dosahuje správných výsledků, a proto by bylo vhodné metodu upravit. Pokud se nám podaří opravit vrstevnice, které leží nejbližší vrcholu, získáme tak vhodná data pro použití metody kolmic na vrstevnice 3.3.3 a šíření výšek mezi jednotlivými vrstevnicemi. Převod na výškovou mapu je závislý na správném okótování vrstevnic. Pokud máme správně okótovány vrstevnice, převod na výškovou mapu 3.4 by nám již neměl činit problémy.

Pokračování této bakalářské práce by mělo spočívat hlavně ve zlepšování dosavadních metod a návrhu nových metod, které by byly více odolnější proti porušeným vrstevnicím. Například bychom mohli z vrcholů postupně „zaplavovat“ okolí, a tak lépe vyhledávat sousední vrstevnice. Dále by bylo vhodné pracovat s více metodami kótování zároveň, jednotlivým metodám přiřazovat určité pravděpodobnosti ohodnocení, ze kterých bychom poté určovali správnou výšku. Vhodné by bylo využít informaci o tom, že každá pátá vrstevnice je na mapě vyznačena širší čarou. Tuto informaci jsem prozatím pro návrh algoritmu nepoužíval, protože dané značení je pouze v některých mapách. Zajímavým rozšířením by byla automatická detekce vrcholů a rozpoznání výšky vrcholu pomocí OCR.

Poslední úpravou, která by zlepšila použitelnost aplikace, je vytvoření intuitivnějšího uživatelského rozhraní. Současné uživatelské rozhraní je postaveno na knihovně OpenCV, která má omezené prostředky pro tvorbu uživatelských rozhraní. Součástí grafického rozhraní by mohly být nástroje pro ruční opravu vrstevnic, protože v některých případech oprava jen jedné přerušené vrstevnice může výrazně zlepšit celkový výsledek.

Rád bych pokračoval ve vývoji této aplikace a vytvořil tak plnohodnotnou a uživatelsky příjemnou aplikaci pro digitalizaci map, která zjednoduší orientaci v mapách.

# Literatura

- [1] Bierza Daniel. *Vytvoření výškové mapy z vrstevnic v rastrové mapě, [bakalářská práce]*. VUT Brno, 2006.
- [2] Václav Hlaváč Milan Šonka. *Počítačové vidění*. Grada a.s., 1992, s. 23, 70, 97.
- [3] Dr. Ing. Eduard Sojka. *Digitální zpracování a analýza obrazů*. VŠB - Technická univerzita Ostrava, 2000, s. 74, 88.
- [4] WWW stránky. Generation 5 - thresholding and segmentation.  
<http://www.generation5.org/content/2003/segmentation.asp> (květen 2008).
- [5] WWW stránky. Gimp - uživatelská příručka.  
<http://docs.gimp.org/2.2/cs/filters-edge.html> (květen 2008).
- [6] WWW stránky. Pov-ray height field.  
<http://www.povray.org/documentation/view/3.6.1/279/> (květen 2008).
- [7] WWW stránky. School of computer science.  
<http://www.cs.cmu.edu/afs/cs/academic/class/15462/web.06s/asst/project1/>  
(květen 2008).
- [8] WWW stránky. Wikipedia - gaussian blur.  
[http://en.wikipedia.org/wiki/Gaussian\\_blur](http://en.wikipedia.org/wiki/Gaussian_blur) (květen 2008).
- [9] WWW stránky. Wikipedia - odstranění šumu.  
[http://cs.wikipedia.org/wiki/Odstranění\\_šumu](http://cs.wikipedia.org/wiki/Odstranění_šumu) (květen 2008).
- [10] WWW stránky. Wikipedia - sobel operator.  
[http://en.wikipedia.org/wiki/Sobel\\_operator](http://en.wikipedia.org/wiki/Sobel_operator) (květen 2008).