

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STAVEBNÁ FAKULTA

MODELOVANIE A VIZUALIZÁCIA
ŠÍRENIA LESNÝCH POŽIAROV

Diplomová práca

BRATISLAVA 2020

Bc. Tatiana Malinaričová

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STAVEBNÁ FAKULTA

Evidenčné číslo: SvF-5343-81237

MODELOVANIE A VIZUALIZÁCIA
ŠÍRENIA LESNÝCH POŽIAROV

Diplomová práca

Študijný program: Matematicko-počítačové modelovanie
Študijný odbor: Matematika
Školiace pracovisko: Katedra matematiky a deskriptívnej geometrie
Vedúci záverečnej práce: Ing. Martin Ambroz, PhD.

BRATISLAVA 2020

Bc. Tatiana Malinaričová



ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Tatiana Malinaričová**
ID študenta: 81237
Študijný program: matematicko-počítačové modelovanie
Študijný odbor: matematika
Vedúci práce: Ing. Martin Ambroz, PhD.

Názov práce: **Modelovanie a vizualizácia šírenia lesných požiarov**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

- Naprogramovanie softvéru pre vizualizáciu modelu šírenia lesných požiarov.
1. Vypracujte prehľad softvérov simulujúcich šírenie lesných požiarov.
 2. Stručne popíšte použitý model šírenia požiaru a jeho implementáciu.
 3. Vytvorte užívateľské prostredie v jazyku C++ s použitím Qt knižnice, v ktorom:
 - je možné voliť vstupné dáta a parametre modelu
 - s použitím navolených parametrov spustiť výpočet pre šírenie požiaru
 - je možné vizualizovať a ukladať časový priebeh vývoja požiaru na palivovej a/alebo terénnej mape v 2D a v 3D

Riešenie zadania práce od: 27. 01. 2020

Dátum odovzdania práce: 14. 05. 2020

Bc. Tatiana Malinaričová
študentka

prof. RNDr. Radko Mesiar, DrSc.
vedúci pracoviska

prof. RNDr. Karol Mikula, DrSc.
garant študijného programu

POKYNY na vypracovanie diplomovej práce

Úvodné ustanovenie

V zmysle zákona č. 131/2002 Z. z. o vysokých školách a o zmene a doplnení niektorých zákonov v znení neskorších predpisov je súčasťou štúdia podľa každého študijného programu aj záverečná práca. Jej obsah patrí medzi štátne skúšky. Záverečnou prácou pri štúdiu podľa študijného programu druhého stupňa je diplomová práca. Podkladom na vypracovanie diplomovej práce je zadanie diplomovej práce

Štruktúra záverečnej práce

- titulný list,
- zadanie záverečnej práce,
- pokyny na vypracovanie,
- vyhlásenie autora,
- názov a abstrakt v slovenskom a v anglickom jazyku (spolu v rozsahu jednej strany),
- obsah s očíslovaním kapitol,
- zoznam príloh,
- zoznam skratiek a značiek,
- text samotnej práce (odporúčané členenie),
 - úvod,
 - súčasný stav problematiky,
 - ciele záverečnej práce,
 - vlastné riešenie členené na kapitoly podľa charakteru práce,
 - zhodnotenie dosiahnutých výsledkov resp. navrhnutých riešení,
 - záver,
- resumé v slovenskom jazyku v rozsahu spravidla 10 % rozsahu ZP (len pre práce vypracované v cudzom jazyku),
- zoznam použitej literatúry,
- prílohy (výkresy, tabuľky, mapy, náčrty) vrátane postera s rozmermi 1000x700 mm.

Rozsah a forma

1. Obsah a forma záverečnej práce musí byť spracovaná v zmysle vyhlášky MŠVVaŠ SR č. 233/2011 Z. z., ktorou sa vykonávajú niektoré ustanovenia zákona č. 131/2002 Z. z. a v zmysle Metodického usmernenia č. 56/2011 o náležitostiach záverečných prác.
2. Vyžadovaný rozsah diplomovej práce je 30 až 50 strán. Odovzdáva sa v dvoch vyhotoveniach. Jedno vyhotovenie musí byť viazané v pevnej väzbe (nie hrebeňovej) tak, aby sa jednotlivé listy nedali vyberať. Rozsiahle grafické prílohy možno v prípade súhlasu vedúceho práce odovzdať v jednom vyhotovení.

3. Autor práce je povinný vložiť prácu v elektronickej forme do akademického informačného systému. Autor zodpovedá za zhodu listinného aj elektronickeho vyhotovenia.
4. Po vložení záverečnej práce do informačného systému, predloží autor fakulte ním podpísaný návrh licenčnej zmluvy. Návrh licenčnej zmluvy je vytvorený akademickým informačným systémom.
5. Odporúčaný typ písma je Times New Roman, veľkosť 12 a je jednotný v celej práci. Odporúčané nastavenie strany - riadkovanie 1,5, okraj vnútorný 3,5 cm, vonkajší 2 cm, zhora a zdola 2,5 cm, orientácia na výšku, formát A4.
6. Obrázky a vzorce sa čísloujú v rámci jednotlivých kapitol (napr. obr. 3.1 je obrázok č. 1 v kapitole 3). Vzorce sa čísloujú na pravom okraji riadku v okrúhlych zátvorkách - napr. (3.1).
7. Všetky výpočty musia byť usporiadané tak, aby bolo možné preveriť ich správnosť.
8. Pri všetkých prevzatých vzorcoch, tabuľkách, citovaných častiach textu musí byť uvedený prameň.
9. Citovanie literatúry vrátane elektronickejších materiálov sa uvádza podľa STN ISO 690 (01 0197): 2012. *Informácie a dokumentácia. Návod na tvorbu bibliografických odkazov na informačné pramene a ich citovanie.*
10. Príklad zoznamu bibliografických odkazov:
 ABELOVIČ, J. a kol.: *Meranie v geodetických sieťach*. Bratislava, Alfa 1990, ISBN 0-1554-9173.
 MICHALČÁK, O. – ADLER, E.: Výskum stability dunajských hrádzí. In: *Zborník vedeckých prác Stavebnej fakulty SVŠT*. Bratislava: Edičné stredisko SVŠT 1976, s. 17-28. ISBN 0-3552-5214.
 ŠÜTTI, J.: Určovanie priestorových posunov stavebných objektov. *Geodetický kartografický obzor*. 2000, roč. 2, č. 3, s. 8-16. ISSN 0811-6900.
 Article 18. Technical Cooperation. <http://www.lac.uk/iso/tc456> (2013-09-28)
11. Za jazykovú a terminologickú správnosť záverečnej práce zodpovedá diplomant.
12. Formu postera (elektronická alebo aj tlačaná) určí garant študijného programu.
13. Vzor pre poster je uvedený na dokumentovom serveri v akademickom informačnom systéme univerzity.

.....
 podpis garanta študijného programu

Ustanovenia týchto pokynov som vzal na vedomie. Som si vedomý(á), že ak nebude moja diplomová práca vypracovaná v súlade s týmito pokynmi, nebude prijatá na obhajobu.

V Bratislave

.....
 podpis študenta

Čestné vyhlásenie

Čestne vyhlasujem, že diplomovú prácu s názvom: "Modelovanie a vizualizácia šírenia lesných požiarov" som vypracovala samostatne, na základe konzultácií a s použitím uvedených informačných zdrojov a literatúry.

V Bratislave, 13.5. 2020

.....
podpis autora práce

Pod'akovanie

Ďakujem predovšetkým môjmu vedúcemu práce Ing. Martinovi Ambrozovi, PhD. za odborné konzultácie, cenné rady a pripomienky, ktoré mi boli prínosom pri písaní diplomovej práce.

Abstrakt

V tejto diplomovej práci sa zameriavame na naprogramovanie užívateľského rozhrania pre model šírenia hranice lesného požiaru. Použitý matematický model je založený na empirických zákonoch šírenia a vstupujú doň vonkajšie vplyvy ako vplyv vetra, sklonu terénu a vplyvy geodetickej a normálovej krivosti krivky. Hodnoty koeficientov týchto vplyvov má možnosť užívateľ sám meniť v užívateľskom prostredí a vytvoriť tak vlastnú simuláciu. Užívateľské prostredie je naprogramované pomocou knižníc Qt a programovacieho jazyka C++. Pomocou novovytvoreného softvéru je možné zobrazenie vývinu krivky spolu s dátami palivovej a terénnej mapy v 2D alebo 3D vizualizácií v rôznych farebných schémach. Výsledky simulácie sa dajú uložiť vo forme obrázkov alebo ako súradnice hranice požiaru vo formáte vtk.

Kľúčové slová: simulátor šírenia lesného požiaru, vývoj krivky, užívateľské rozhranie

Abstract

In this diploma thesis, we focus on the development of the user interface for a forest fire spread simulator. Implemented mathematical model is based on empirical laws of fire spread. Its input parameters are external forces such as wind influence, terrain slope influence, geodesic and normal curvatures of the curve. The user is able to change the values of these parameters and to create his own simulation. The user interface is developed by using the Qt libraries and the programming language C++. With the help of the newly created software, it is possible to display the evolution of the curve together with the data from fuel and terrain maps in 2D or 3D in several color schemes. The simulation results can be saved as an image or as curve coordinates representing fire perimeter in vtk format.

Keywords: forest fire spread simulator, curve evolution, user interface

Obsah

1	Úvod	3
2	Prehľad softvérov simulujúcich šírenie lesných požiarov	4
2.1	Simulátory založené na fyzikálnom modelovaní	4
2.1.1	Wildland-urban interface Fire Dynamics Simulator	4
2.2	Porovnanie simulátorov založených na vektorovom a rastrovom prístupe modelovania	6
2.2.1	FARSITE	6
2.2.2	Vesta	8
3	Implementovaný matematický model šírenia požiaru	9
3.1	Popis modelu vývoja krivky	10
3.2	Implementácia modelu	12
4	Užívateľské prostredie	15
4.1	Popis grafického rozhrania	18
4.1.1	Rozloženie - Layouty	20
4.2	Triedy a objekty	21
4.2.1	Marching squares algoritmus	23
4.2.2	Farebné schémy	28
4.2.3	Import a export súborov vo formáte VTK	29
4.2.4	3D vizualizácia dát	30
5	Praktická ukážka simulácie	33
6	Záver	38

1 Úvod

Požiare v prírode a straty lesov spojené s nimi sú v súčasnosti vážnym problémom a stále aktuálnejšou témou. Základné podmienky pre spustenie požiaru sú iskra a palivo, ale pri dnešných extrémnych klimatických podmienkach v niektorých miestach ako je sucho a vysoké teploty je spustenie požiaru o to pravdepodobnejšie a požiar je o to intenzívnejší. V posledných desaťročiach sa priemerné teploty stále viac zvyšujú a s nimi aj vznik požiarov po celom svete. Najviac ohrozované miesta sú na území južnej Európy, Severnej Ameriky a jeden z najväčších požiarov bol teraz na území Austrálie. S požiarimi takýchto veľkých rozmerov okrem iného prichádza aj hrozba vyhynutia niektorých živočíšnych druhov. Preto je dôležité predchádzať takýmto stratám tým, že budeme vedieť predvídať správanie a šírenie ohňa. S tým nám pomáhajú rôzne druhy simulátorov šírenia požiarov.

Simulátory šírenia požiarov modelujú správanie požiaru, teda ako sa šíri naprieč krajinou s rôznorodým palivom, klímou a topografiou terénu. Využití simulátorov je mnoho, či už na simulácie potenciálneho požiaru, lokalizovanie požiaru alebo taktickú podporu pri aktívnych požiaroch a rekonštrukcií požiaru.

Cieľom tejto práce je naprogramovanie užívateľského prostredia na vizualizáciu výsledkov modelu šírenia požiaru pre rôzne modelové situácie a rôzne vstupné dáta.

V druhej kapitole je spracovaný prehľad druhov simulačných softvérov a popis pár konkrétnych príkladov.

Tretia kapitola je zameraná na popis a implementáciu matematického modelu, ktorý bol vytvorený na pôde Katedry matematiky a deskriptívnej geometrie STU. Konkrétne ide o Lagrangeovský model, teda krivka je reprezentovaná diskretnými bodmi, ktoré sa pohybujú v závislosti od viacerých vstupných faktorov.

V štvrtej kapitole sa dostaneme ku cieľu tejto diplomovej práce, teda samotnej vizualizácii šírenia požiaru a návrhu užívateľského prostredia pre model šírenia lesných požiarov, jeho implementácií a funkcionalite. Vstupom od užívateľa sú nutné dáta, ako mapa horľavosti paliva v teréne, mapa sklonu terénu, smer a rýchlosť vetra a počiatočná krivka.

V piatej kapitole sme sa zamerali na ukážku konkrétneho príkladu s rôznymi vstupnými parametrami.

2 Prehľad softvérov simulujúcich šírenie lesných požiarov

Modelovanie lesných požiarov je náročné z dôvodu rôznych podmienok prostredia (počasie, terén, vegetácia) a rozsahu trvania fyzikálnych procesov riadenia, akými sú vznietenie, šírenie ohňa a dymu. Úplné riešenie problému šírenia lesných požiarov si vyžaduje model, ktorý je rýchlejší ako samotný požiar v reálnom čase. Takýto model je schopný v čase nehody zachytiť dôležité procesy šírenia a operatívne tak pomôcť zasahujúcim jednotkám. Ak by mal takýto model podrobne popisovať fyzikálne procesy, ako napr. toky tepla, vznikli by priveľké nároky na rýchlosť procesora a pamäť počítača. Naopak, v situácií, keď nejde o výpočtový čas, je možné klásť dôraz aj na detaily procesu šírenia. Aj preto existuje viacero typov modelov. [3]

Je viacero spôsobov, ktorými sa dá pristupovať k modelovaniu vývinu hranice požiaru. Simulátory preto môžeme rozdeliť napríklad na fyzikálne a empirické. Empirické simulátory sú založené zvyčajne na štatistickej analýze a pozorovaniach šírenia požiaru. Ďalším spôsobom delenia môže byť, že pracujú buď na rastrovom (Eulerovom), alebo vektorovom (Lagrangeovom) princípe. Pri rastrovom prístupe je pozícia novej krivky počítaná v každej bunke mriežky, zatiaľ čo diskkrétne body novej krivky šírenia požiaru pri vektorovom prístupe sú počítané v každom časovom kroku.

2.1 Simulátory založené na fyzikálnom modelovaní

Dvojrozmerné modely šírenia ohňa založené na fyzikálnom modelovaní využívajú zákony zachovania. Dominantným spôsobom prenosu tepla sú radiácia ako hlavný systém prenosu tepla a prúdenie, ktoré reprezentuje účinok vetra a sklonu terénu. Takýto model vedie k riešeniu systému Navier-Stokesových parciálnych diferenciálnych rovníc.

Vo všeobecnosti modely založené na fyzikálnych princípoch vyžadujú dlhší výpočtový čas oproti empirickým.

2.1.1 Wildland-urban interface Fire Dynamics Simulator

Wildland-urban interface Fire Dynamics Simulator (WFDS) [3] je rozšírením softvéru Fire Dynamic Simulator (FDS). FDS je voľne dostupný softvér vytvorený National Ins-

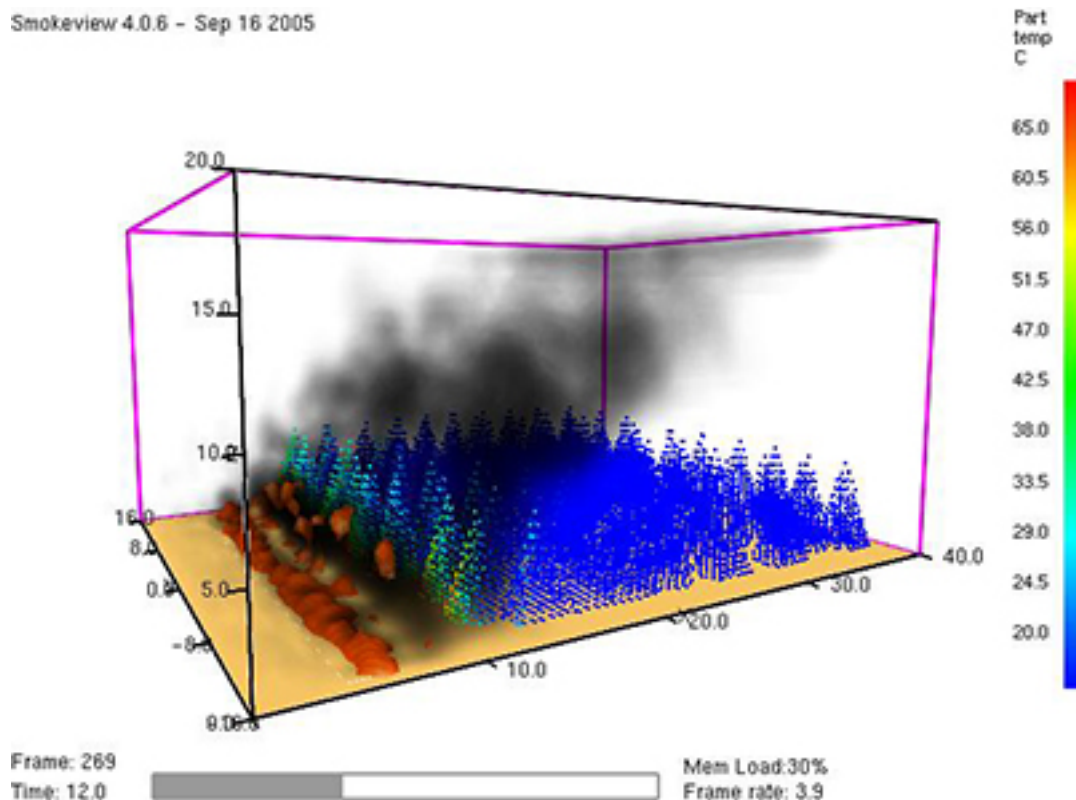
tute of Standards and Technology (NIST) Ministerstva obchodu Spojených štátov v spolupráci s Fínskym strediskom technického výskumu. Smokeview je sprievodný vizualizačný program, ktorý možno použiť na zobrazenie výstupu FDS.

WFDS tvoria v skutočnosti 2 modely: WFDS-PB, založený na fyzikálnom modelovaní a WFDS-LS, semi-empirický model, kde je šírenie krivky ohňa založené na modelovaní vrstevníc. Rozdiel medzi nimi je len vo vstupných údajoch.

Vstupnými údajmi pre WFDS je napríklad veľkosť územia, počet uzlových bodov, špecifikácia spaľovania palivových plynov, prítoková rýchlosť, okrajové podmienky územia, časové intervaly, v ktorých sa zapisujú výstupné dáta a výstup pre Smokeview. Príklad výstupu softvéru Smokeview je zobrazený na obrázku 1.

Na modelovanie vegetácie sú vytvorené dve metódy: metóda palivových elementov a metóda okrajového paliva. Prvá metóda môže byť použitá na reprezentáciu povrchu alebo zvýšenej vegetácie a metóda okrajového paliva sa používa len na reprezentáciu povrchového paliva a modeluje vegetáciu na hrubšej sieti uzlových bodov ako prvá metóda.

Vstupnými dátami pre metódu palivových elementov sú súbory, ktoré špecifikujú bližšie vegetáciu, termofyzikálne vlastnosti, pozícia a geometria vegetácie.



Obr. 1: Príklad výstupu softvéru Smokeview z [3]

2.2 Porovnanie simulátorov založených na vektorovom a rastrovom prístupe modelovania

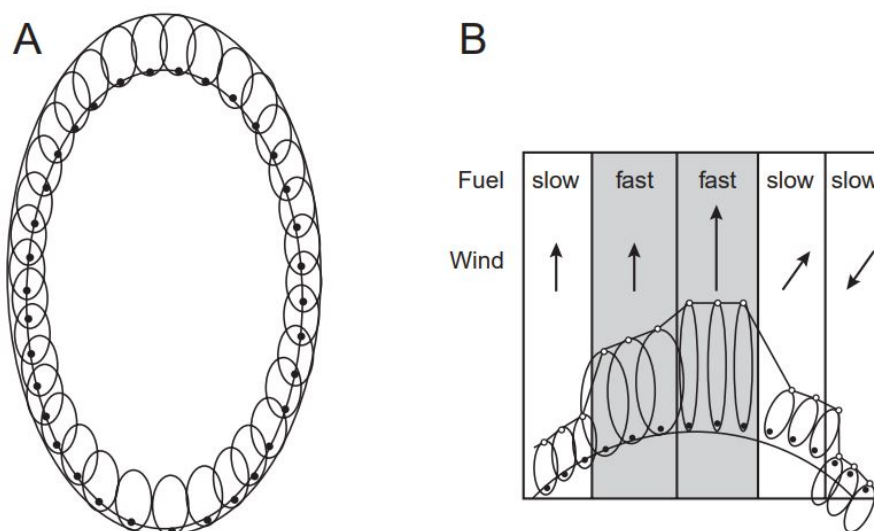
Vo všeobecnosti majú algoritmy založené na rastrových modeloch menšie nároky na procesor pre dosiahnutie porovnateľných výsledkov s vektorovými modelmi. Pri vektorových modeloch treba ošetriť situácie, kedy môže nastať samopriesek jednej krivky, zjednotenie viacerých izolovaných kriviek atď. Neošetrenie takýchto situácií vedie k nesprávnym výsledkom a program sa stáva nestabilným. Práve riešenie týchto topologických zmien pomerne výrazne zaťažuje a spomaľuje väčšinu vektorových modelov.

2.2.1 FARSITE

Fire Area Simulator (FARSITE) [1] je predstaviteľom empirických simulátorov z kategórie vektorových a jeden z najpoužívanejších programov na simuláciu šírenia požiaru, ktorý využíva už vytvorené modely správania požiarov. Zahŕňa modely šírenia požiaru (Rothermelov model), šírenia požiaru koruny (Van Wagnerov model), bodové zrýchlenie požiaru a vlhkosť paliva (Nelsonov model). Je používaný štátnymi a fe-

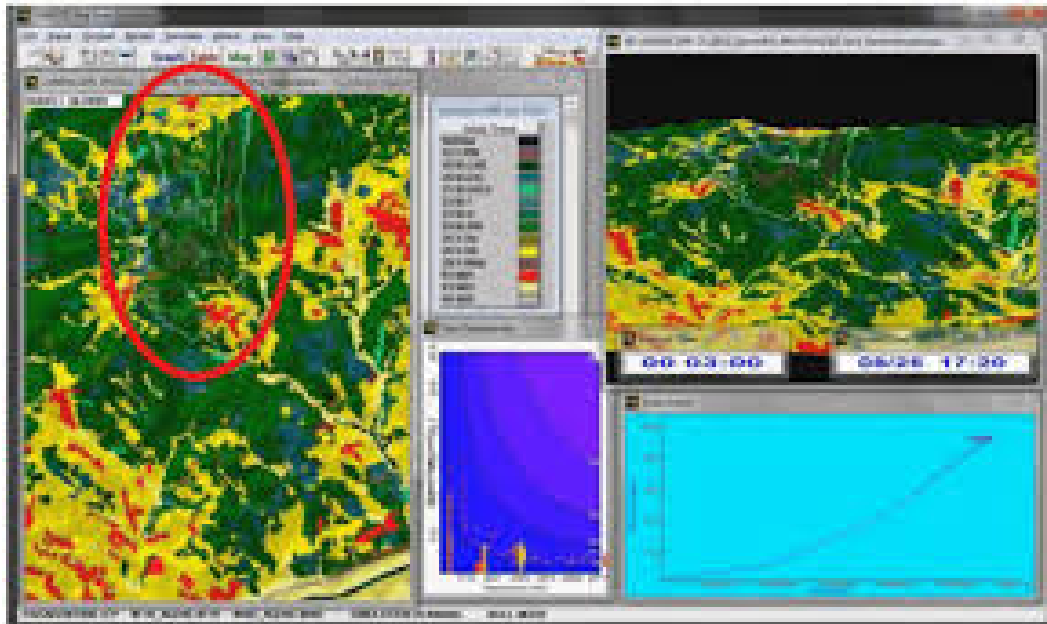
derálnymi agentúrami v USA.

Je príkladom modelu založenom na vektorovom prístupe modelovania šírenia požiaru, kde využíva Huygensov princíp. Krivka sa šíri ako neustále sa zväčšujúci polygón v konkrétnom časovom kroku. Polygón je definovaný 2D vrcholmi a ich počet sa zväčšuje s narastajúcim časom. Hlavnou myšlienkou Huygensovho princípu aplikovaného na šírenie požiaru (Richardsov model) je, že každý z týchto vrcholov je zdrojom ďalšieho nezávislého šírenia, teda ďalšej elipsy, čo je vidieť z obr. 2. Tvar a smer elipsy sú dané vektorom vetra, zatiaľ čo veľkosť je určená rýchlosťou šírenia a dĺžkou časového intervalu.



Obr. 2: Huygensov princíp s použitím eliptických vln [1]. (A) Rovnomerné podmienky, elipsy sú rovnakej veľkosti v každom časovom kroku. (B) Nerovnomerné podmienky, vidieť závislosť veľkosti vln od horľavosti paliva a smeru a rýchlosti vetru.

Užívateľské rozhranie [2] na obr. 3 bolo vyvinuté pre operačný systém Microsoft Windows. FARSITE model vyžaduje od užívateľa na vstup súbory s dátami o teréne, palive, klíme a vetre. Následne pomocou myši užívateľ určí ohnisko na načítanej mape terénu. Ohnisko môže zadať ako body, čiary alebo už existujúce tvary. Užívateľ má taktiež možnosť robiť malé zmeny na teréne, určiť druh paliva, trvanie simulácie, teda koncový čas a uložiť si výstupné dáta.



Obr. 3: Uživatelské prostredie softvéru FARSITE

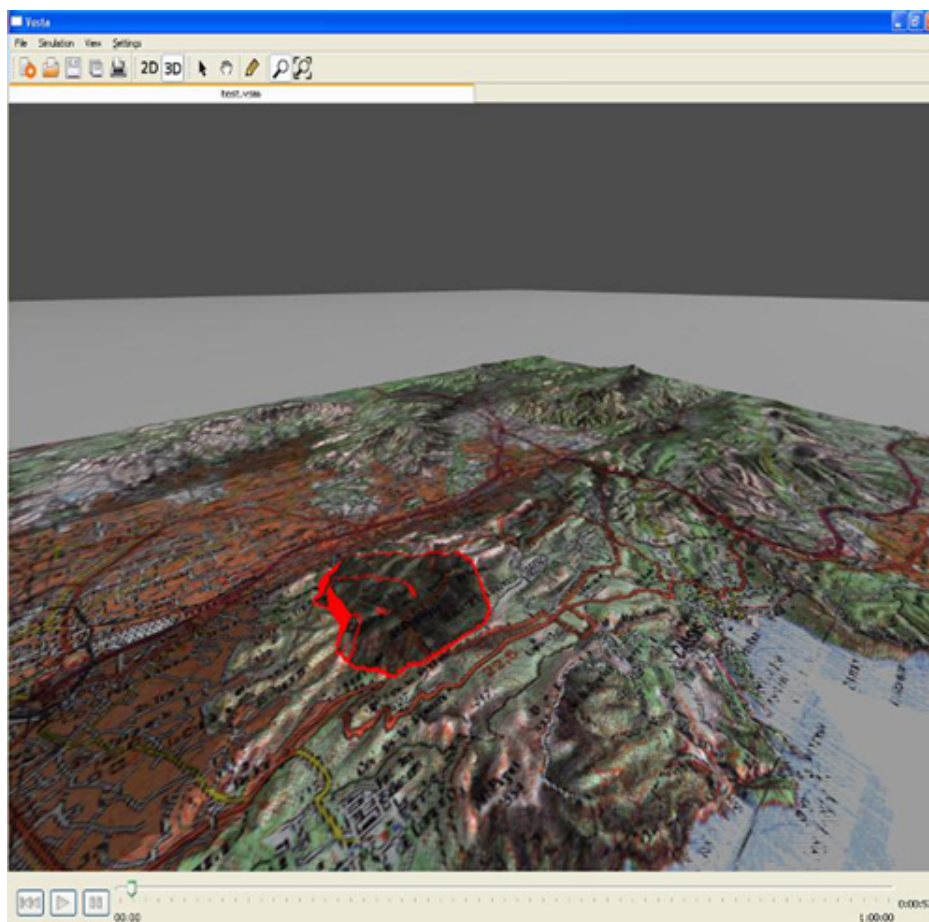
2.2.2 Vesta

Vesta je simulátor založený na rastrovom princípe a adaptovaný pre rozsiahle požiare v Európe, založený na fyzikálnom princípe. Je vyvinutý francúzskym tímom MTDA, ako súčasť programu European Fire Paradox [4].

Program Vesta rozdeľuje simuláciu šírenia požiaru na dva problémy:

- model požiaru - počíta rýchlosť šírenia v ktoromkoľvek bode územia, závislú od sklonu terénu, smeru a rýchlosti vetra, vegetácie a podobne
- mechanizmus požiaru šírenia - ako sa požiar šíri medzi jednotlivými bunkami mriežky

Uživatelské prostredie simulátoru Vesta je zobrazené na obr. 4.



Obr. 4: Uživatelské prostredie programu Vesta z [4].

3 Implementovaný matematický model šírenia požiaru

Model šírenia krivky z [5] je vytvorený na základe Lagrangeovského modelu, teda krivka, reprezentovaná bodmi sa pohybuje v závislosti od rôznych vplyvov prostredia a parametrov, ktoré si zvolí užívateľ. Tieto vplyvy sú v modeli vyjadrené pomocou empirických zákonov.

V tejto kapitole sú naznačené základné princípy aplikované v modeli a popísané jednotlivé vplyvy (empirické zákony) na smer a rýchlosť šírenia požiaru. Ďalej sú v kapitole popísané vstupné dáta, ich formát a využitie a na záver samotná implementácia modelu do programu.

3.1 Popis modelu vývoja krivky

Hranicu požiaru šíriaceho sa v teréne reprezentuje krivka \mathcal{G} na ploche \mathcal{M} . Plocha \mathcal{M} je dvojrozmerná plocha v \mathbb{R}^3 , $\mathcal{M} = \{(x, y, \varphi(x, y)) \in \mathbb{R}^3, (x, y) \in \Omega\}$, reprezentovaná grafom funkcie φ na oblasti $\Omega \subset \mathbb{R}^2$, pričom φ reprezentuje digitálny model terénu. Pre numerické výpočty sa využíva priemet do roviny, čím z krivky \mathcal{G} dostaneme rovinnú krivku Γ .

Rovinná krivka Γ sa pohybuje v čase podľa rovinného rýchlostného vektorového poľa. Pohyb každého bodu \mathbf{x} krivky Γ môžeme rozdeliť do dvoch smerov, tangenciálneho a normálového a vyjadriť podľa vzťahu

$$\mathbf{x}_t = \beta \mathbf{N} + \alpha \mathbf{T}, \quad (1)$$

kde β predstavuje rýchlosť v smere jednotkovej vonkajšej normály \mathbf{N} , α v smere jednotkovej tangenty \mathbf{T} rovinnej krivky Γ .

Aplikovaním rovníc pre geodetickú a normálovú krivosť, normálovú rýchlosť krivky povrchu a využitím Frenet - Seretových vzťahov $\mathbf{x}_{ss} = -k\mathbf{N}$ z [5] môže byť rovnica 1 prepísaná do formy intrinsickej parciálnej diferenciálnej rovnice pre polohový vektor \mathbf{x} krivky Γ

$$\mathbf{x}_t = \varepsilon \mathbf{x}_{ss} + \alpha x_s + w x_s^\perp, \quad (2)$$

kde ε predstavuje krivostný člen, α advekčný člen a w je silový člen.

Kvôli numerickému riešeniu je rovnica 2 diskretizovaná, krivostný člen semi - implicitnou schémou a advekčný člen inflow-implicit/outflow-explicit - nou metódou, respektíve implicitnou upwind schémou.

Vplyvy geodetickej a normálovej krivosti. Tvar obvodu krivky, teda jej krivosť taktiež ovplyvňuje vývin požiaru. Geodetická krivosť predstavuje vplyv lokálneho tvaru krivky, teda krivku vyhladzuje. Normálová krivosť krivky lokálne znižuje alebo zvyšuje normálovú rýchlosť v závislosti od tvaru topografie. V prípade doliny je normálová krivosť kladná, teda lokálna normálová rýchlosť sa zvýši. Naopak na hrebene kopca je normálová krivosť záporná, takže lokálna normálová rýchlosť sa bude znižovať. Takže vývin krivky môže byť popísaný rýchlosťou ν v smere vonkajšej normály

$$\nu = F(\delta_F - \delta_g \kappa_g + \delta_n \kappa_n), \quad (3)$$

kde F je vonkajšia sila, δ_F predstavuje vplyv vonkajšej sily, δ_g a δ_n sú vplyvy geodetickej

a normálovej krivosti, κ_g a κ_n predstavujú geodetickú a normálovú krivosť. Vzťah pre vonkajšiu silu je založený na empirických zákonoch šírenia požiaru v prírode, teda na vplyve paliva, sklonu terénu a počasí, teda vplyvu vetra

$$F = f f_w(\mathbf{w} \cdot \mathcal{N}) f_s(\mathbf{s} \cdot \mathcal{N}), \quad (4)$$

kde f je vplyv paliva, f_w je vplyv vetra, f_s je vplyv sklonu terénu, \mathbf{w} je trojrozmerný vektor vetra, \mathcal{N} je jednotková vonkajšia normála krivky \mathcal{G} .

Vplyv paliva. Šírenie požiaru závisí od vlastností paliva, ktoré je väčšinou heterogénne. Palivová mapa teda predstavuje skalárnu funkciu $f(x)$, danú kombináciou faktorov ako sú hustota, vek a druh lesa, pričom mladé husté ihličnaté lesy šíria požiar najrýchlejšie.

Vplyv vetra. Vietor má vplyv na rýchlosť a smer šírenia požiaru. Môže znížiť alebo zvýšiť rýchlosť šírenia požiaru, v závislosti od toho, či prúdi rovnakým alebo opačným smerom. Podporuje horenie, tým že mu dodáva kyslík.

Šírenie ohňa je ovplyvnené vetrom exponenciálne, takže vzťah pre funkciu vplyvu vetra vyzerá nasledovne

$$f_w(\mathbf{w} \cdot \mathcal{N}) = e^{\lambda_w(\mathbf{w} \cdot \mathcal{N})}, \quad (5)$$

kde λ_w je kladné číslo. Od skalárneho súčinu $(\mathbf{w} \cdot \mathcal{N})$ závisí vplyv vetra na funkciu šírenia požiaru:

- $\mathbf{w} \cdot \mathcal{N} = 0 \Rightarrow f_w = 1$, vektory sú na seba kolmé a funkcia šírenia nie je vetrom nijako ovplyvnená,
- $\mathbf{w} \cdot \mathcal{N} = |\mathbf{w}| \Rightarrow$ vektory majú rovnaký smer a vplyv vetra je najsilnejší,
- vo zvyšných prípadoch je exponent f_w daný ako projekcia vektoru vetra \mathbf{w} na vonkajšiu normálu \mathcal{N} .

Vplyv sklonu terénu Podobne ako v predchádzajúcom prípade, vplyv sklonu terénu na funkciu šírenia požiaru je exponenciálny a rýchlosť šírenia môže zvýšiť, znížiť alebo zmeniť jej smer. Vzťah pre funkciu vplyvu sklonu terénu vyzerá nasledovne

$$f_s(\mathbf{s} \cdot \mathcal{N}) = e^{\lambda_s(\mathbf{s} \cdot \mathcal{N})}, \quad (6)$$

kde λ_s je kladné číslo.

Faktor tangenciálnej redistribúcie. Kvôli numerickej stabilite výpočtu je na body krivky aplikovaná tangenciálna redistribúcia, čo je nápomocné aj pri prípadoch rozdelenia krivky alebo zlúčenia viacerých do jednej.

3.2 Implementácia modelu

Do nášho programu sme implementovali matematický model šírenia požiaru pomocou Dll knižnice. Dynamic-link library (Dll), teda dynamicky spojené knižnice sú knižnice používané operačným softvérom Windows a sú implementáciou zdieľaných knižníc. Dll môže obsahovať triedy, funkcie, premenné, obrázky, externé zdroje, užívateľské rozhranie a podobne. Rozdiel medzi statickou a dynamickou knižnicou je, že statické knižnice sú pripojené k spúšťanému súboru v čase kompilácie, zatiaľ čo dynamické až pri spustení. To okrem iného znamená aj to, že v prípade potreby môžeme aktualizovať používanú dll knižnicu bez nutnosti kompilácie programu, ktorý ju používa.

Nutnými vstupmi od užívateľa na spustenie simulácie sú nasledovné mapy a parametre, ktoré ovplyvňujú vývoj krivky požiaru [5]:

- Palivová mapa
- Mapa terénu
- Dĺžka časového kroku
- Počet časových krokov
- Maximálna normálová rýchlosť
- Vplyv paliva
- Vplyv geodetickej krivosti
- Vplyv normálovej krivosti
- Vplyv vetra
- Vplyv sklonu terénu
- Faktor tangenciálnej redistribúciej
- Dĺžka jedného segmentu krivky

- Rýchlosť a smer vetra
- Počiatočné krivky

Palivová mapa je greyscale obrázok typu *.bmp*. Stupeň šedi predstavuje lokálnu rýchlosť šírenia požiaru bez akéhokoľvek vonkajšieho vplyvu, či už vetra, sklonu terénu a podobne. Čierna farba predstavuje nulovú rýchlosť, teda nehorľavé časti, akými sú napríklad rieka alebo cesta, biela farba reprezentuje maximálnu rýchlosť, teda najviac horľavé časti (mladé husté lesy). Dáta z palivovej mapy sú ukladané pre ďalšie použitie v rozsahu 0 - 1, namiesto celočíselného rozsahu 0 - 255.

Mapa terénu je načítavaná ako textový súbor, ktorého hlavička je vo formáte: počet riadkov, počet stĺpcov, priestorový krok v x a y rovnaký a nadmorské výšky v danom bode siete. V programe je reprezentovaná rôznymi farebnými schémami, napríklad stupňami šedej, kde stupeň šedi predstavuje nadmorskú výšku, pričom biela je najvyšší bod a čierna najnižší alebo pomocou vrstevníc.

Počiatočná krivka. Vstupom do metódy na výpočet vývinu krivky je vektor súradníc bodov počiatočnej krivky alebo kriviek reprezentujúcich ohnisko vzniku požiaru alebo model krivky požiaru z predošlého časového kroku. Počiatočné krivky môžu byť zadané ručne užívateľom alebo načítané z VTK súboru.

Hodnoty všetkých vstupných parametrov majú východzie nastavenia. Koeficienty vplyvov vetra, paliva, geodetickej a normálovej krivosti sú nastavené na hodnotu 1. Zvolené môžu byť v intervale 0-1.

Počas celého priebehu simulácie je vietor konštantný, reprezentovný 2D vektorom. Užívateľ ho zadá pomocou "kompasu", rýchlosť reprezentuje dĺžka vektoru a smer pozícia vektoru na "kompase". Tieto dáta môžu byť poskytnuté meteorologickými stanicami.

Na výpočet vývoja krivky sa používa veľmi malý časový krok, približne 0.1 minúty. Ak sa požiar pohybuje 1m/min, tak pri väčších požiaroch je tento pohyb veľmi ťažko pozorovateľný. Takže, aby sa predišlo načítaniu zbytočne veľkého množstva kriviek, tak užívateľ ďalej musí zadať, či chce vykreslenie každého desiateho alebo stého časového kroku vývoja krivky. Funkcia teda vráti krivku po zvolenom počte časových krokov, tá sa uloží do výsledného vektora a funkcia sa opäť zavolá s novou počiatočnou krivkou.

Vo funkciách sa najprv vytvoria objekty pre triedy vstupných parametrov, mapy paliva a mapy terénu. Do konštruktora objektu triedy vstupných parametrov ide veľkosť

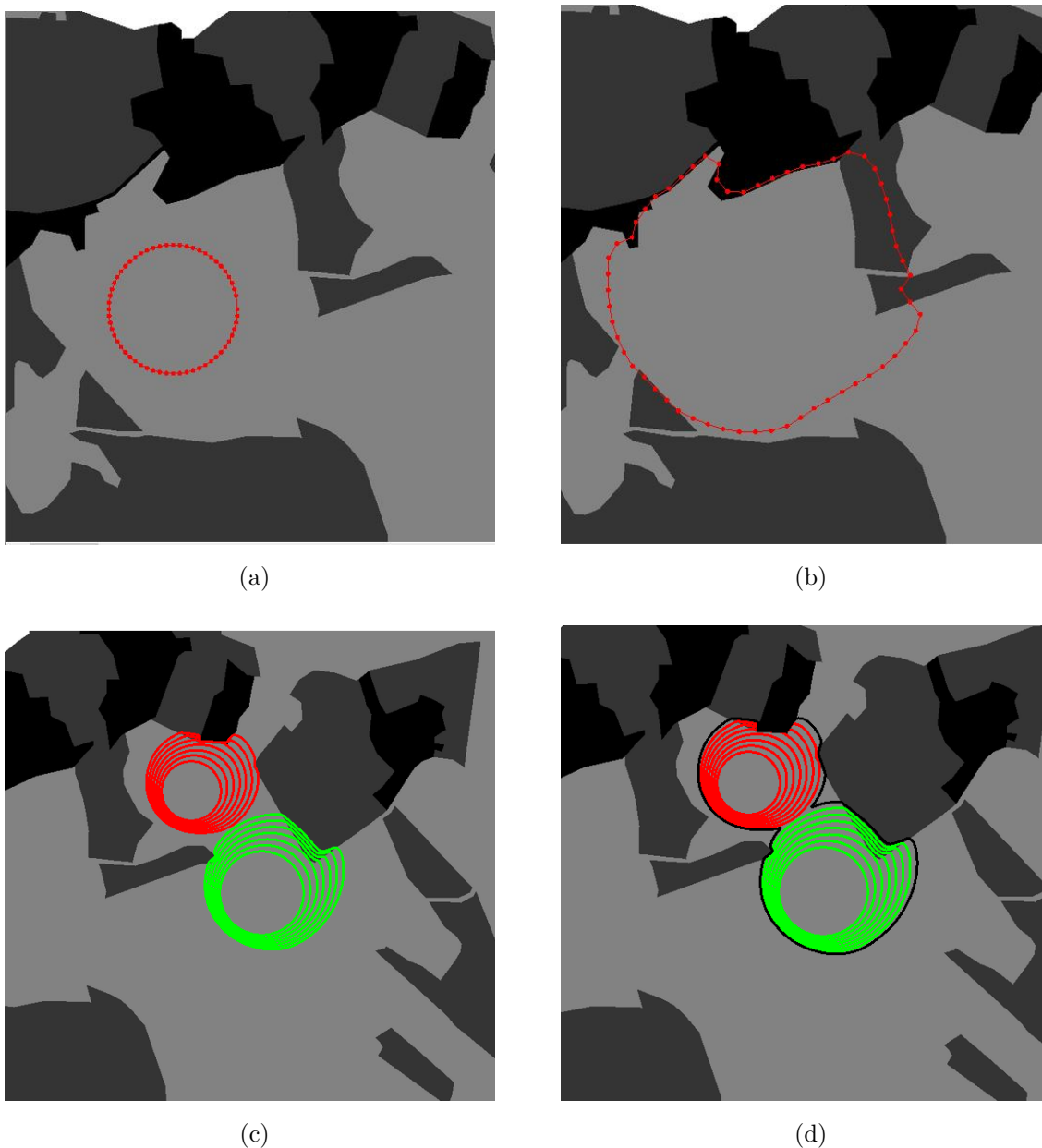
časového kroku, časový krok pre zápis krivky, koeficient tangenciálnej redistribúcie, koeficient vplyvu geodetickej krivosti, vplyv vetra, vplyv sklonu terénu, vektor smeru vetra a dĺžka jedného segmentu krivky.

Konštruktor triedy mapy paliva potrebuje na vstup dáta palivovej mapy, šírku a výšku mapy, priestorový krok siete mapy a maximálnu normálovú rýchlosť.

Do konšuktora triedy mapy terénu ide vektor mapy gradientov terénu, hesiánová matica, teda matica druhých derivácií terénu počítaná už z vypočítaného gradientného vektoru, počet riadkov a počet stĺpcov terénnej mapy a priestorový krok v x a y smere.

Samotná funkcia na výpočet vývinu sa volá v cykle n -krát, kde n je pomer celkového počtu časových krokov a časového kroku pre zápis a jej vstupom je vektor súradníc počiatočnej krivky a objekty vyššie spomínaných vytvorených tried.

Návratovou hodnotou tejto metódy je vektor súradníc bodov všetkých kriviek, ktoré vznikli vývinom tej pôvodnej vo všetkých časových krokoch. Vizualizáciu vývinu bodov krivky je vidieť na [5](#).



Obr. 5: (a) je počiatková podmienka, (b) je vývin bodov krivky po tisíc časových krokoch, na (c) a (d) zobrazujú vývin krivky vo všetkých časových krokoch. Čierna krivka na (d) predstavuje spojenie červenej a zelenej krivky.

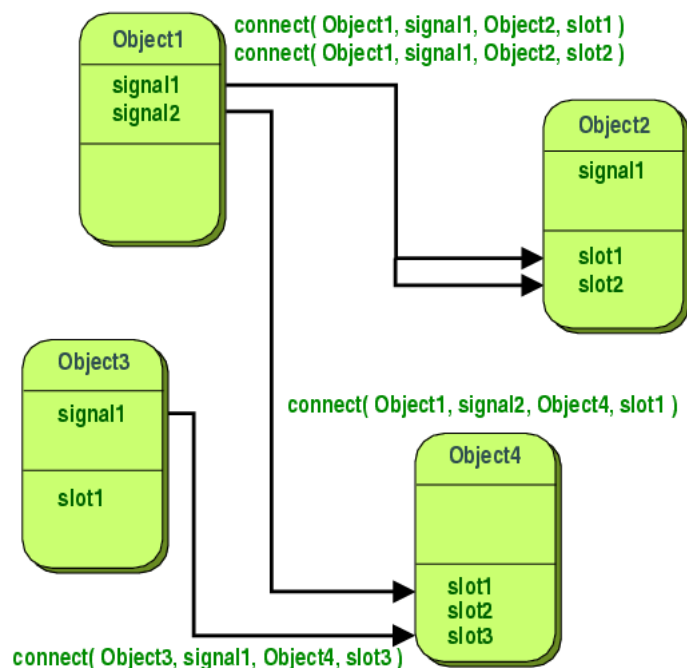
4 Uživatelské prostredie

Uživatelské prostredie 7 je naprogramované v jazyku C++ s použitím knižníc Qt [6]. Qt je jedna z najpopulárnejších multiplatformových knižníc pre vytváranie programov s grafickým užívateľským rozhraním. Qt toolkit bol vytvorený v roku 1999 spoločnosťou Trolltech, ktorá ho v roku 2008 predala firme Nokia. Od roku 1999 sa Qt toolkit vyvinul

na multiplatformový nástroj, v ktorom možno vyvíjať konzolové alebo GUI aplikácie v odlišných programovacích jazykoch pre rôzne platformy.

Qt nie je samostatný programovací jazyk, je to knižnica, ktorú je možné implementovať vo viacerých programovacích jazykoch, vrátane C++. Na rozšírenie jazyka C++ o funkcie ako sú signály a sloty sa používa preprocesor MOC (Meta - Object Compiler). MOC číta hlavičkový súbor. Ak nájde jedno alebo viac deklarácií tried, ktoré obsahujú makro Q_OBJECT, vytvorí zdrojový súbor C++ obsahujúci kód metaobjektov pre tieto triedy.

Významnou funkcionalitou Qt sú signály a sloty. Slúžia na komunikáciu medzi objektami. Ak sa uskutočnila nejaká akcia, vyšle sa signál a ako odpoveď sa zavolá príslušná funkcia, slot. Qt má preddefinované množstvo signálov aj slotov, ale je možné vytvárať vlastné. Napríklad, keď klikneme na tlačidlo Zatvoriť, okno sa zavrie. Pri prepájaní signálov a slotov, môže na jeden signál byť napojených viacero slotov a naopak jedným slotom môže byť spojených viacero signálov, viď obr. 6.

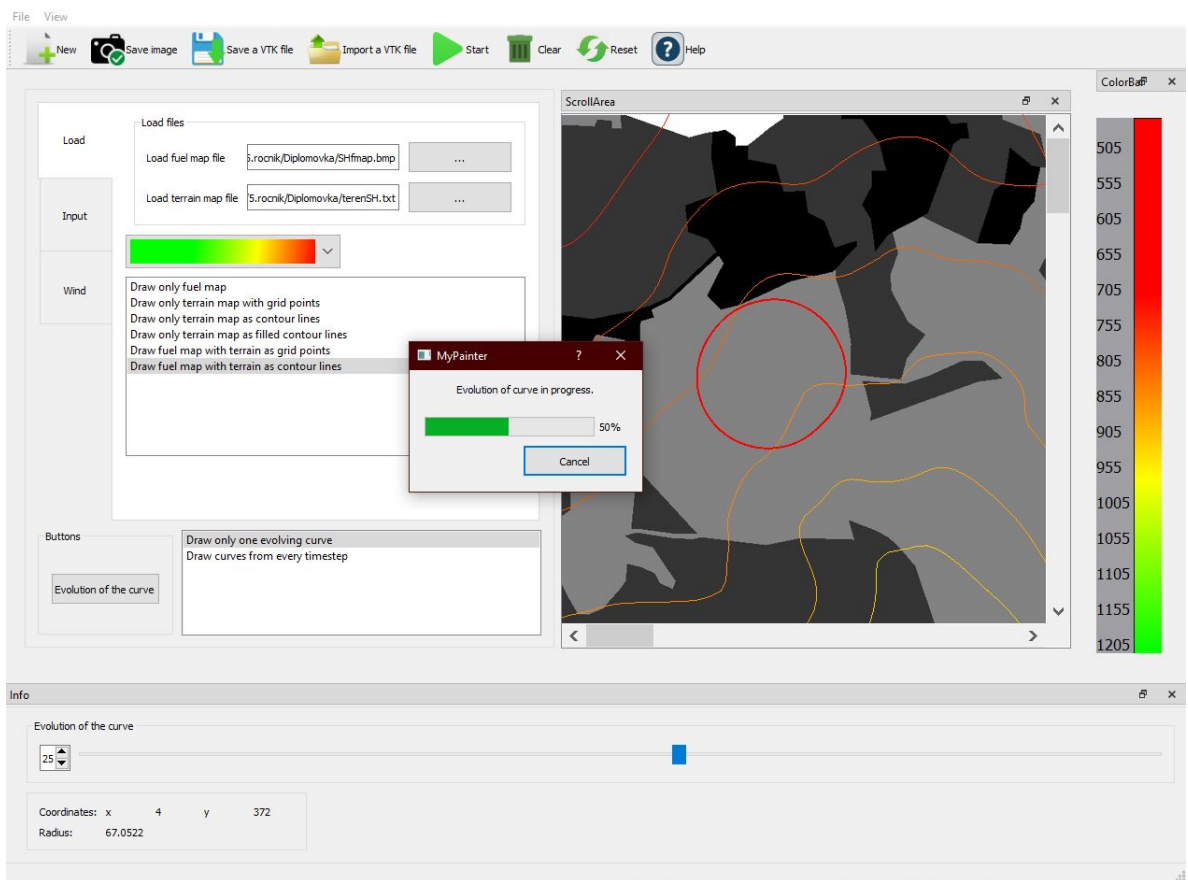


Obr. 6: Príklad prepojenia signálov a slotov.[6]

Konkrétny príklad, ktorý demonštruje to, keď posunieme slider "slider_posun" o jeden krok, tak sa vyšle signál "valueChanged(int)" a pomocou slotu "EvoDraw(int)" sa vykreslí krivka v príslušnom časovom kroku:

```
connect(slider_posun, SIGNAL(valueChanged(int)), this, SLOT(EvoDraw(int)));
```

Grafické užívateľské rozhranie môže byť vytvorené buď dynamicky priamo v C++ s použitím Widgetov (element GUI, ktorý zobrazuje alebo odovzdáva informácie pomocou interakcie s užívateľom), alebo grafickým nástrojom Qt Designer. Qt Designer je interaktívny grafický nástroj, ktorý funguje ako generátor kódu pre grafické rozhrania založené na Widgetoch. Naše grafické rozhranie na obr. 7 je naprogramované dynamicky. Dôraz bol kladený na funkcionality a vizualizáciu šírenia modelu požiaru v 2D aj v 3D.



Obr. 7: Užívateľské rozhranie.

Lagrangeovou metódou sa zo zadaných vstupných parametrov vypočíta určený počet časových krokov šírenia krivky. Model je schopný pracovať s viacerými nezávislými počiatočnými ohniskami požiaru a ošetrené sú aj zmeny kriviek, respektíve hranice požiaru v závislosti od topologických zmien, teda spojenie dvoch individuálnych požiarov dohromady alebo jeden rozdeliť na viacero. V práci [5] je prezentovaný algoritmus na detekovanie a spracovanie takýchto zmien s časovou náročnosťou $O(n)$, kde n je počet bodov krivky.

4.1 Popis grafického rozhrania

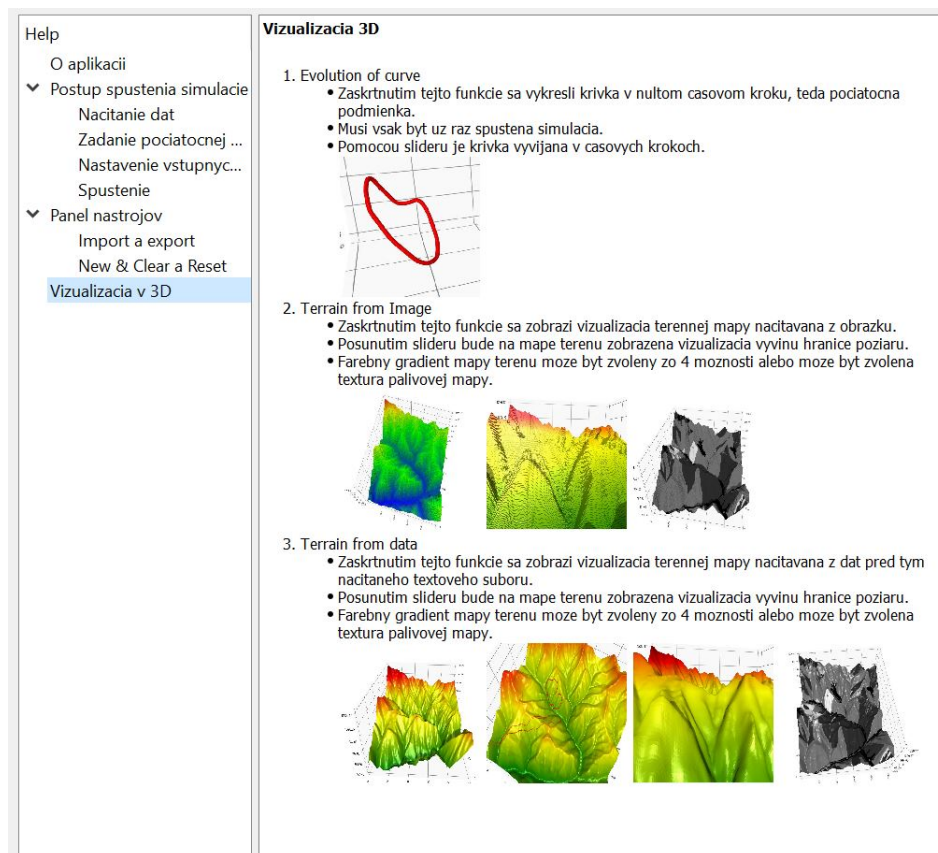
Grafické rozhranie je vytvorené dynamicky pomocou Widgetov, bez využitia Qt Designera. Okno aplikácie, obr. 7, pozostáva z vrchnej lišty, panelu nástrojov a hlavného okna rozdeleného na dve časti - záložky pre vstupné údaje a kresliace plátno.

Vo vrchnej lište sa nachádzajú rolovacie možnosti File a View.

V paneli nástrojov, obr. 8, sú ikony, ktoré odkazujú na funkcie pre uloženie obrázku z kresliaceho plátna, uloženie krivky do VTK súboru, načítanie krivky z VTK súboru, spustenie výpočtu vývinu krivky, zresetovanie alebo vymazanie kresliaceho plátna a help, obr. 9. Help slúži ako stručný návod popisujúci funkcionality aplikácie. Obsahuje krátku charakteristiku simulátora, podkapitolu, ako postupovať pri spustení simulácie a stručný popis funkcií panelu nástrojov.



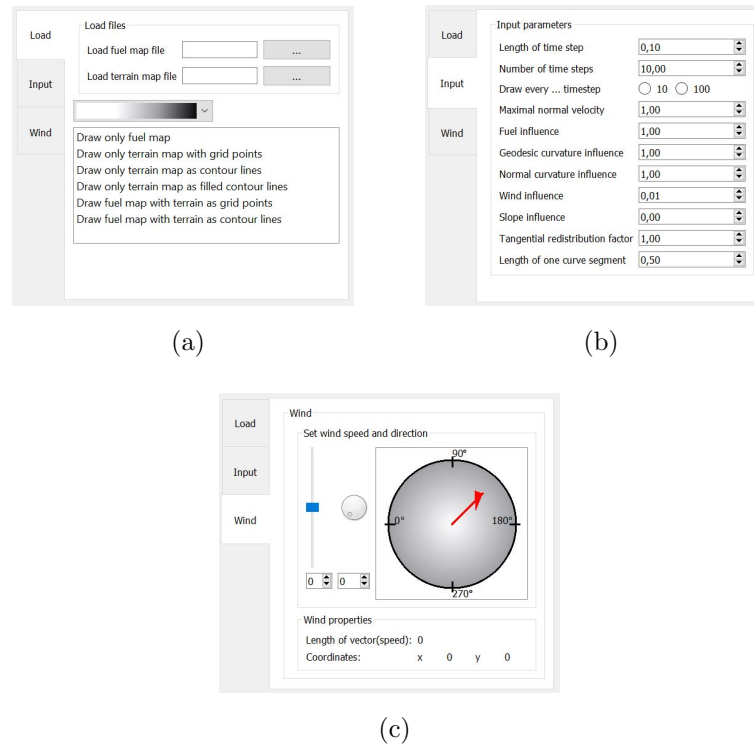
Obr. 8: Panel nástrojov z užívateľského rozhrania.



Obr. 9: Okno Helpu, konkrétne zobrazená je časť o 3D vizualizáciách.

Hlavná časť okna je rozdelená na dve časti, ktorých pomer veľkosti si môže užívateľ

podľa potreby meniť. Na ľavej strane sú umiestnené tri záložky pre vstupné údaje. Jednotlivé záložky sú zobrazené na nasledujúcich obrázkoch 10:



Obr. 10: (a) záložka pre načítanie a vykreslenie súborov, (b) záložka pre zadanie hodnôt vstupných koeficientov a na obrázku (c) sa nachádza "kompas", pre určenie smeru a rýchlosti vetra

Na pravej strane je kresliace plátno, v ktorom sa zobrazia súbory pre palivovú mapu, mapu terénu a samotný vývoj krivky na týchto mapách. Pod kresliacim plátnom sa nachádza informačný panel s posúvačom, obr. 11, slúžiacim na posúvanie medzi jednotlivými časovými krokmi pre vizualizáciu vývoja krivky a výpis aktuálnych súradníc kurzora na mape a polomer počiatocnej kružnice.



Obr. 11: Informačný panel z užívateľského rozhrania. Vo vrchnej časti spinBox prepojený s posúvačom slúžia na prepínanie časových krokov vývinu krivky, v spodnej časti informácia o aktuálnych súradniciach kurzora na mape a polomer počiatočného ohniska.

4.1.1 Rozloženie - Layouty

Systém rozloženia v Qt je založený na princípe layoutov, ktorý poskytuje jednoduchý spôsob automatického usporiadania Widgetov, aby sa predišlo hluchým miestam a dobre sa využil priestor. Qt obsahuje viacerov typov rozloženia (Form Layout, Vertical Layout, Grid Layout...) widgetov vrámci rozhrania aplikácie. Ak sa zmení veľkosť využiteľného priestoru, tak layouty automaticky menia veľkosť a umiestnenia widgetov a tým zabezpečujú konzistentnosť rozhrania.

Rozloženie v našej aplikácii je nasledovné. Rozhranie aplikácie je rozdelené na dve hlavné časti. Jedna časť slúži na zadávanie vstupných parametrov a načítanie súborov a druhá na zobrazovanie dát.

Prostredie zadávania potrebných vstupných dát a ich úprava je vytvorené pomocou triedy `QTabWidget`. Z vizuálneho hľadiska je to prehľadný spôsob, ako zobrazíť a zoradiť väčšie množstvo vstupných parametrov. Ďalší spôsob pridávania widgetov do rozhrania je pomocou triedy `QDockWidget`. Táto trieda poskytuje widgety, ktoré môžu byť ukotvené v hlavnom okne alebo neukotvené, teda sú o úroveň vyššie ako hlavné okno. Pomocou tejto triedy je vytvorený informačný panel, ktorý zobrazuje aktuálne súradnice, polomer zadávaného ohniska a posúvač pre vývin krivky, grafické plátno a farebná stupnica.

Vo vrchnej časti hlavného okna sa nachádza aj panel nástrojov vytvorený pomocou triedy `QToolBar` s funkciami na uloženie kresliacej plochy vo formáte `.png`, uloženie a importovanie kriviek vo formáte `VTK`, vymazanie všetkých načítaných dát alebo ich zresetovanie, spustenie výpočtu evolúcie kriviek a help.

4.2 Triedy a objekty

Trieda v C++ je základným budovacím prostriedkom. Je to dátový typ definovaný užívateľom, ktorý obsahuje atribúty a metódy (funkcie). K atribútom a metódam triedy pristupujeme cez inštanciu danej triedy. Jej deklarácia je v hlavičkovom súbore *.h a definícia v súbore *.cpp.

Objekt je inštancia triedy, teda je to premenná dátového typu triedy. Pri definícii triedy sa nealokuje žiadna pamäť, alokuje sa až pri vytvorení jej inštancie.

V našom programe je využitých množstvo tried, väčšina sú už preddefinované triedy Qt knižníc. Novovytvorené triedy:

- FireSimulator
- paintWidget
- LevelCurves
- TabProxy
- ColorBar
- Circle
- Wind
- Surface3D

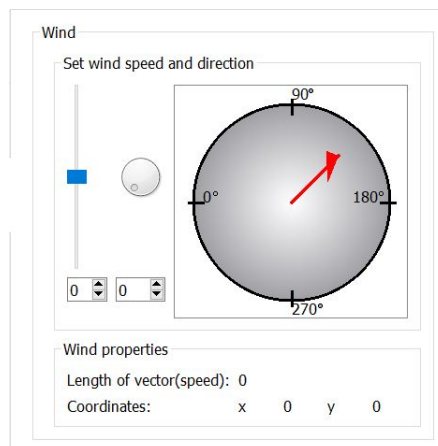
Hlavnou triedou je *FireSimulator*, ktorá obsahuje objekty widgetov, objekty zvyšných vyššie uvedených tried, metódy na volanie funkcií ostatných tried a metódy na dynamické vytváranie grafického prostredia.

Ďalšou triedou je *paintWidget*, ktorá obsahuje metódy, ktorých výstup je viditeľný na kresliacej ploche. Napríklad metódy na načítanie, zobrazenie a uloženie súborov dát máp a kriviek. Import a export kriviek vo formáte VTK je bližšie popísaný v kapitole [4.2.3](#).

Metódami triedy *LevelCurves* dostávame ako návratovú hodnotu súradnice výškových vrstevníc terénu podľa algoritmu popísaného v kapitole [4.2.1](#).

TabProxy je trieda na vytvorenie záložkového widgetu v transponovanej podobe.

Na obrázku 12 je znázornená implementácia triedy *Wind*. Trieda obsahuje funkcie na vykreslenie "kompasu", kde pomocou vektoru v podobe šípky je možné určiť veľkosť a smer vetra, buď uchopením myšou alebo pomocou widgetov. Súradnice bodov hrotu šípky sú prepočítavané pomocou polárnych súradníc. Smerový vektor je návratovou hodnotou jednej z funkcií, ktorá ide na vstup do konštruktora objektu vstupných hodnôt v metóde pre vývoj hranice požiaru.



Obr. 12: Layout s funkciami na zadanie parametrov vetra. Vektor v kompase sa dá nastaviť buď uchopením šípky myšou alebo pomocou widgetov na ľavej strane. V spodnej časti je výpis aktuálnej veľkosti vektoru a aktuálnych súradníc polohy hrotu šípky.

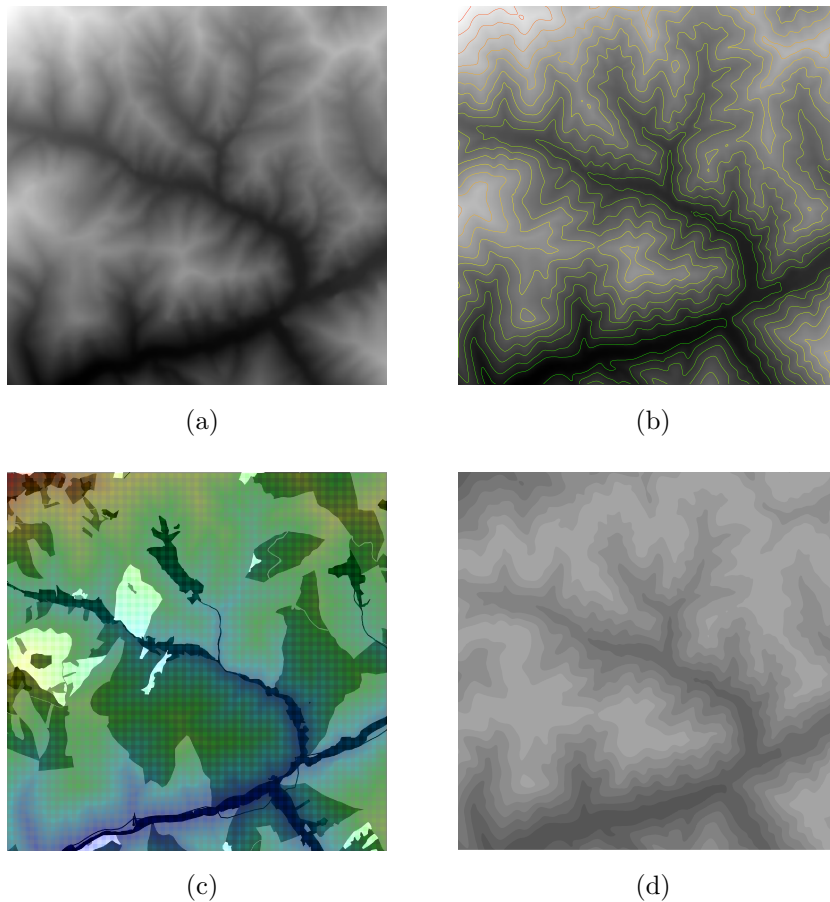
Metódy triedy *Circle* vrátia vypočítané body počiatočného ohniska požiaru v podobe kružnice a jej vykreslenie.

Trieda *ColorBar* obsahuje iba metódu na vrátenie príslušnej farebnej škály, kde je využitá trieda *QLinearGradient*. Užívateľ má na výber zo štyroch farebných škál, ktorými si vie zobrazit' terénnu mapu rovnako ako palivovú. Užívateľ má možnosť zobrazit' si zvlášť palivovú mapu s rôznou farebnou schémou, zvlášť mapu terénu s rôznou farebnou schémou a dokopy palivovú mapu v grayscale schéme spolu s vrstevnicami nadmorskej výšky v ľubovolnej farebnej škále. Farebným schémam je venovaná kapitola 4.2.2.

Zobraziť terén spolu s vývojom hranice požiaru v 3D umožňujú metódy triedy *Surface3D*, bližšie popísaná v kapitole 4.2.4. Trieda okrem funkcií na načítanie a zobrazenie dát disponuje funkciami na výber farebného gradientu zobrazenia výškovej mapy terénu alebo textúry danej palivovou mapou.

4.2.1 Marching squares algoritmus

Na zobrazenie terénu sú implementované dve možnosti. Terén môže byť zobrazený buď pomocou bodov vo zvolenej farebnej škále v závislosti od nadmorskej výšky, alebo pomocou vrstevníc, izočiar, vid' 25. Rôzne zobrazenia terénu vidieť na obrázku 13. Na konštrukciu izočiar existuje viacero algoritmov, v našej aplikácii bol zvolený algoritmus Marching squares [7]. Algoritmus je naprogramovaný v triede *LevelCurves*.



Obr. 13: Spôsoby zobrazenia terénnej mapy. (a) Samotné zobrazenie pixelov v závislosti od výšky. (b) Obrázok (a) spolu s vrstevnicami. (c) Palivová mapa spolu s výškovými pixelmi. (d) Vyplnené kontúry vrstevníc.

Marching squares algoritmus je analogickým odvodením z algoritmu pre tri dimenzie Marching Cubes, ktorý bol vytvorený v roku 1987 Lorensenom a Clineom.

Hlavnou myšlienkou algoritmu je, že izočiaru môžu byť konštruované po častiach pre každý prvok siete a nezávisle od ostatných častí siete. Algoritmus generuje aproximáciu pre vrstevnicu dvojrozmerného skalárneho poľa. Teda pre 2D funkciu nájde

aproximáciu čiary, kde všetky body na čiare majú rovnakú funkčnú hodnotu. V našom prípade ide o funkciu

$$f(x, y) = h, \quad (7)$$

kde h predstavuje nadmorskú výšku vrstevnice v bode so súradnicami (x, y) .

Výhodou je časová náročnosť, ktorá je rovná počtu štvorcov v sieti.

Algoritmus

Vstupom pre algoritmus je hodnota hľadanej izočiary a vektor hodnôt v uzloch siete. Teda pre náš prípad je vstupom hodnota hľadanej nadmorskej výšky a dáta hodnôt nadmorských výšok z celého terénu. Výstupom algoritmu je n súradníc uzlov izočiary. Algoritmus pozostáva z troch krokov:

1. Zostaviť rovnomernú štvorcovú sieť
2. Priradiť index každému štvorcu siete.
3. Lineárna interpolácia súradníc izočiary.

1. Zostaviť rovnomernú štvorcovú sieť

Sieť je načítaná z dátového súboru mapy terénu "DataTerrain" s daným krokom a uzlom štvorca siete sú priradené hodnoty dát z tohto súboru nasledovne:

$a = \text{DataTerrain}[(i + 1) * \text{length_of_row} + j];$

$b = \text{DataTerrain}[(i + 1) * \text{length_of_row} + (j + 1)];$

$c = \text{DataTerrain}[i * \text{length_of_row} + (j + 1)];$

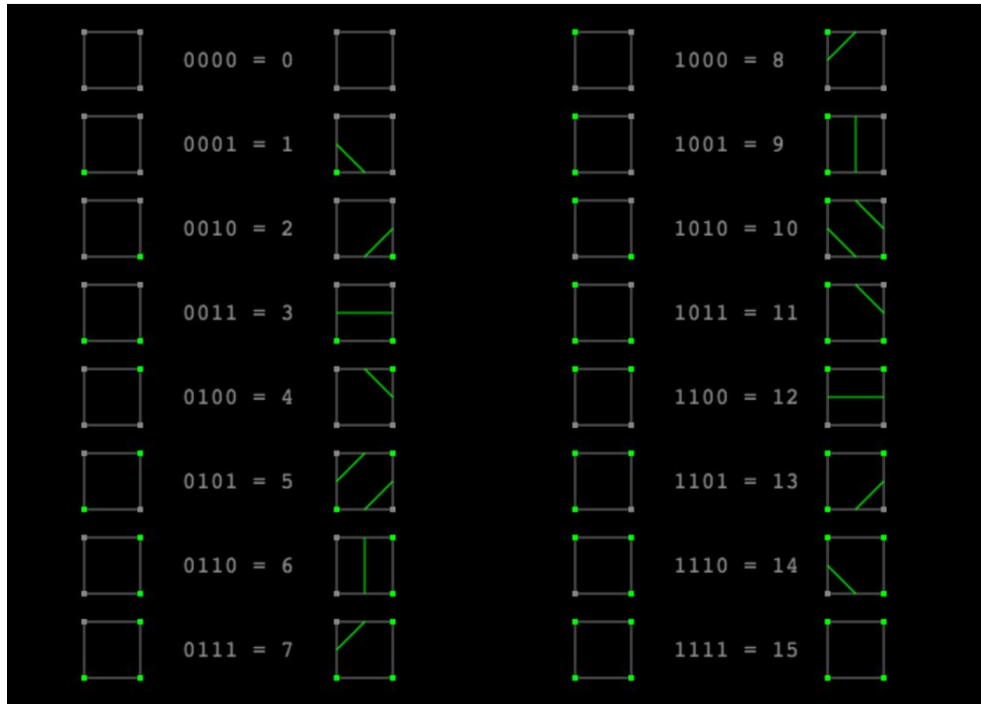
$d = \text{DataTerrain}[i * \text{length_of_row} + j];$

kde hodnoty a, b, c, d predstavujú uzly daného štvorca siete.

2. Priradiť index každému štvorcu siete

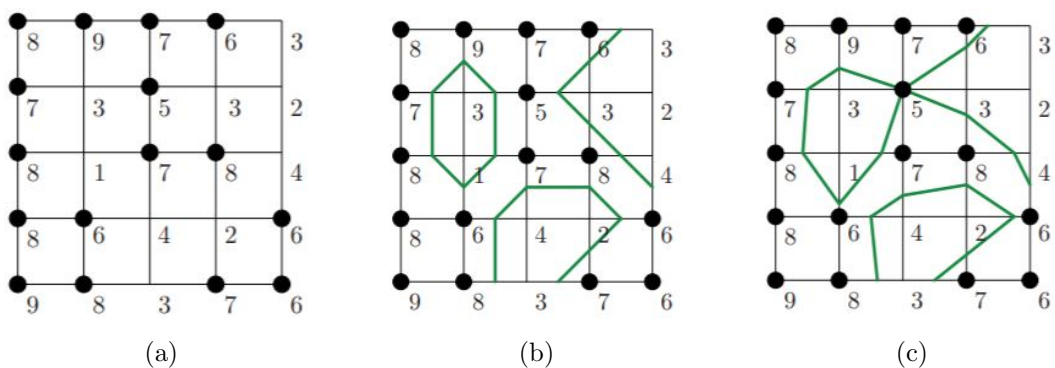
Ďalším krokom je určiť index každého štvorca siete. Každý uzol je buď negatívny alebo pozitívny. Ak je funkčná hodnota uzlu (a, b, c alebo d) $< h$, uzlu je priradená hodnota 0 a uzol je negatívny. V opačnom prípade je uzlu priradená hodnota 1 a uzol je pozitívny. Hodnoty v uzloch konkrétneho štvorca sú interpretované ako binárne číslo, pričom uzly sú zoradené v poradí: ľavý dolný, pravý dolný, pravý horný a ľavý horný ako je vidieť aj na obrázku [14](#).

V každom štvorci sú 4 uzly a teda výsledný index bunky má celkovo $2^4 = 16$ možností usporiadania stavov na uzloch. Podľa hodnoty indexu bunky, bude mať hrana reprezentujúca bunku jeden z nasledujúcich 16 stavov na obrázku 14.



Obr. 14: Konfigurácia vo štvorci, pričom zelený uzol je pozitívny, bitová interpretácia uzlov a k nim príslušná poloha izočiarey. [8]

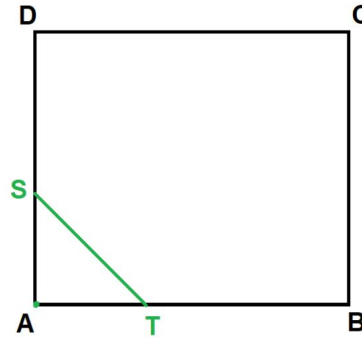
Izočiara pretína tie hrany štvorca, ktoré majú jeden uzol pozitívny a jeden negatívny. Takto izočiarou oddelíme pozitívne uzly od negatívnych, viď 15.



Obr. 15: Príklad štvorcovej siete (a), pri hodnote izočiarey $h=5$. Uzly označené čiernym krúžkom sú pozitívne. Na (b) izočiarey podľa konfigurácie štvorcov pred lineárnou interpoláciou a v (c) po lineárnej interpolácii. [7]

3. Lineárna interpolácia súradníc izočariy

Aby sme určili presnejšie prienik izočariy s hranou, využijeme lineárnu interpoláciu s hodnotami v uzlových bodoch štvorca.



Obr. 16: Príklad jedného štvorca siete, iba uzol A je pozitívny a zvyšok sú negatívne.

Napríklad situácia na obrázku 16 znázorňuje prípad, keď iba v ľavom dolnom rohu je hodnota uzlu väčšia, nanajvýš rovná hodnote izočariy a lineárnou interpoláciou vypočítame aproximáciu súradníc bodov S a T .

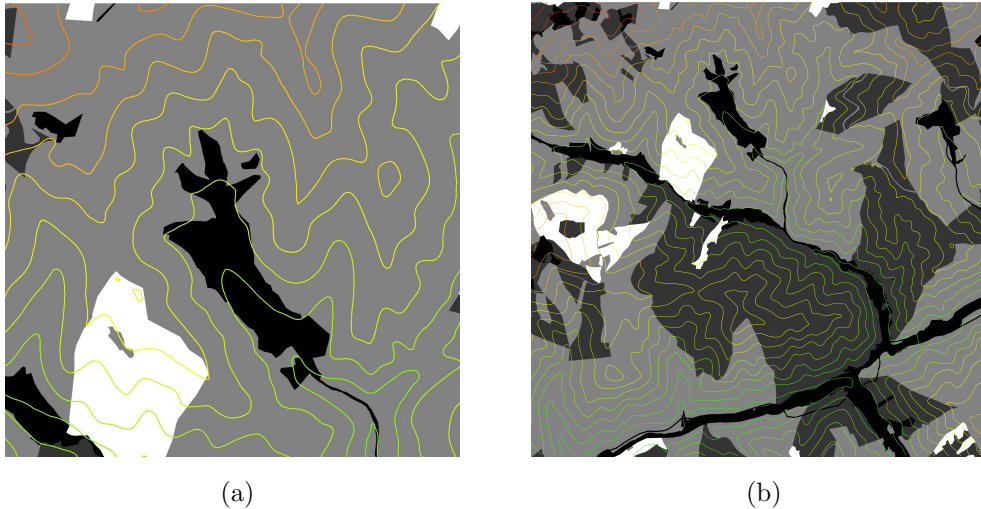
Z obrázku 16 je vidieť, že súradnica x bodov S a D sa rovná, teda $Dx = Sx$ a pre súradnicu Sy platí

$$\frac{Sy - Dy}{Ay - Dy} \approx \frac{f(Sx, Sy) - f(Dx, Dy)}{f(Ax, Ay) - f(Dx, Dy)} \quad (8)$$

odkiaľ vyjadríme Sy . Keďže súradnice izočariy ležia na hrane štvorca, tak bod S nahradíme $f(Sx, Sy) \approx h$ a výsledný vzťah pre Sy bude vyzerat' nasledovne

$$Sy = \frac{h - f(Dx, Dy)}{f(Ax, Ay) - f(Dx, Dy)}(Ay - Dy) + Dy, \quad (9)$$

kde $Ay - Dy$ je hodnota priestorového kroku medzi súradnicami, čo je dané zo súboru a nikdy nemôže byť nulová. Obdobný postup je aplikovaný na zvyšné prípady.

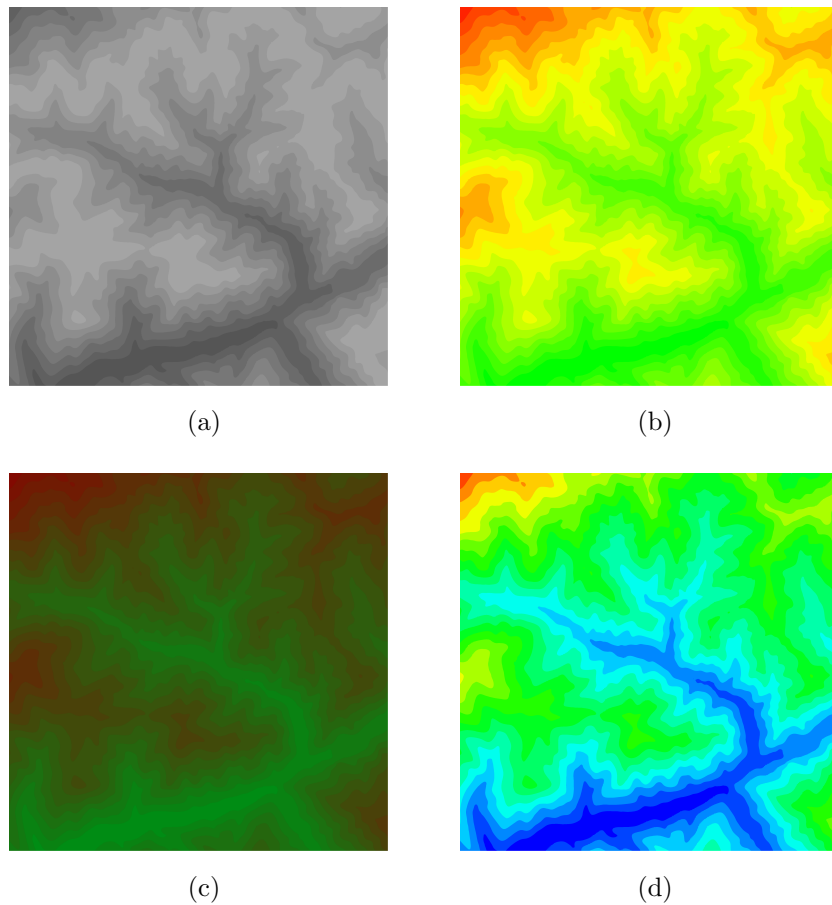


Obr. 17: Na (a) je priblížený výsek palivovej mapy a na (b) celá spolu s vrstevnicami nadmorskej výšky vypočítanými algoritmom Marching squares [7].

Časová náročnosť

Algoritmus beží v lineárnom čase, a ak je počet uzlov v sieti N , tak časová náročnosť algoritmu je $\theta(N)$. Počet uzlov v sieti sa rovná počtu štvorcov v sieti, teda N a každý z nich sa prejde len raz. Detekovanie všetkých hrán izočiary teda zaberie $O(N)$ času. Keďže na každej hrane môže byť maximálny počet bodov izočiary len jeden, tak izočiara má maximálne toľko bodov, koľko má sieť hrán a tých je menej ako dvakrát hodnota uzlov siete, teda aj počet bodov izočiary je menej ako $2N$, čo znamená, že aj výpočtový čas bude lineárny $O(N)$.

Túto metódu sme ešte rozšírili o možnosť vykresliť vrstevnice v podobe "pruhov", v ktorých je priestor medzi vrstevnicami je vyplnený farbou, vid' obr. 18. Do triedy bola pridaná ďalšia funkcia, ktorej návratová hodnota je vektor súradníc bodov výsledného polygónu, ktoré podľa hodnoty indexu elementu určia ako bude výsledný polygón vyzerať. Algoritmus funguje úplne rovnako ako v predošlom prípade. Vo výsledku sú vypočítané polygóny vyplnené farbou podľa užívateľom vybranej farebnej schémy. Na vyplnenie farbou je využitá trieda knižníc Qt *QPainterPath*. Objektu tejto triedy je priradený polygón a metódou *fillPath* sa vyplní príslušnou farbou.



Obr. 18: Na obrázkoch je možné vidieť 4 farebnými schémami znázornenú mapu terénu pomocou vyplnených vrstevníc algoritmom Marching squares.

4.2.2 Farebné schémy

Užívateľ má na výber zo 4 farebných schém ako je vidieť z obrázku 18. Na vypočítanie hodnoty farby pixelu sú vytvorené metódy, ktorých návratová hodnota sú zložky farby r , g , b .

Schéma, kde je prechod len medzi dvoma farbami (červená, zelená) je počítaná lineárnou interpoláciou

$$\begin{aligned}
 color_{red} &= (color_{red}^2 - color_{red}^1) * ratio + color_{red}^1 \\
 color_{green} &= (color_{green}^2 - color_{green}^1) * ratio + color_{green}^1, \\
 color_{blue} &= (color_{blue}^2 - color_{blue}^1) * ratio + color_{blue}^1
 \end{aligned} \tag{10}$$

kde $color^1$ a $color^2$ sú hraničné farby, medzi ktorými sa interpoluje, red , $green$, $blue$ sú zložky týchto farieb a $ratio$ je vstupná hodnota pixelu z obrázku.

Farebné schémy, ktoré sú tvorené prechodom viacerých farieb sú generované jednoduchým algoritmom na základe toho, koľko farieb bude schému tvoriť. Vstupom je hodnota pixelu z intervalu 0 - 1. Táto hodnota je vynásobená počtom prechodov medzi farbami a dĺžkou jedného farebného regiónu, teda 256, nazveme ju m . Následne hodnota, ktorá nám vyjde ako zvyšok po delení m číslom 256, predstavuje vzdialenosť od najbližšieho farebného prechodu x . Ak m predelíme 256, zistíme v ktorom farebnom regióne sa pixel nachádza a na základe toho a x zistíme príslušné hodnoty r, g, b pixelu. Napríklad, ak sa pixel nachádza v regióne žltej, výstupom bude: $r = 255 - x$, $g = 255$, $b = 0$.

4.2.3 Import a export súborov vo formáte VTK

Panel nástrojov ponúka možnosť vloženia a uloženia kriviek vo formáte VTK (*Visualisation Toolkit*). To znamená, že počiatočné ohnisko môže byť aj načítaná krivka zo súboru a krivka vypočítaná v ľubovoľnom časovom kroku môže byť uložená.

VTK je objektovo orientovaná knižnica pre spracovanie a vizualizáciu dát, ktorá poskytuje rôzne funkcie pre manipuláciu s týmito dátami, ako napríklad kontúrovanie, triangulácie, zjemnenie siete atď.

```
# vtk DataFile Version 2.0
testovanie
ASCII
DATASET POLYDATA

POINTS 13 float
0 0 0
1 0 0
1 1 0
0 1 0
10 0 0
11 0 0
11 1 0
10 1 0
12 0.5 0
10 10 0
11 10 0
11 11 0
10 11 0

POLYGONS 3 16
4 0 1 2 3
5 4 5 6 7 8
4 9 10 11 12

CELL_DATA 3
```

Obr. 19: Príklad formátu VTK súboru.

Krivky sú importované a exportované v nasledujúcom formáte (viď aj ukážku na obr. 19):

- Prvý riadok súboru hovorí o verzii VTK.
- Druhý riadok tvorí hlavičku súboru. Pozostáva zo znakov typu *string* a zakončený je znakom `\n`. Hlavička popisuje dáta, čo zobrazujú môžu sa tam o nich nachádzať ďalšie prídavné informácie.
- Ďalší riadok určuje formát súboru, buď ASCII alebo BINARY.
- Štvrtá časť popisuje typ štruktúry dát. Prvá časť musí byť vždy *DATASET* a potom nasleduje typ popisujúci geometriu. V našom prípade *POLYDATA*. Ostatné typy môžu byť napríklad *STRUCTURED_GRID*, *FIELD*...
- Nasleduje posledná časť, ktorá popisuje atribúty dát. Pre náš konkrétny príklad časť *POINTS 13 float* hovorí o tom, koľko bodov a akého formátu tvoria krivky, následne sú vypísané ich súradnice v poradí x, y, z, v tomto prípade vidieť, že krivky sú 2D.
- V riadku *POLYGONS 3 16*, prvé číslo vyjadruje počet polygónov, ktoré tvoria vyššie uvedené body a druhé je súčet počtu polygónov a bodov. Nasledovné riadky popisujú ktorými bodmi sú dané polygóny tvorené. Prvé číslo predstavuje počet bodov a zvyšné sú indexy vyššie uvedených bodov v poradí akom idú v polygóne za sebou.

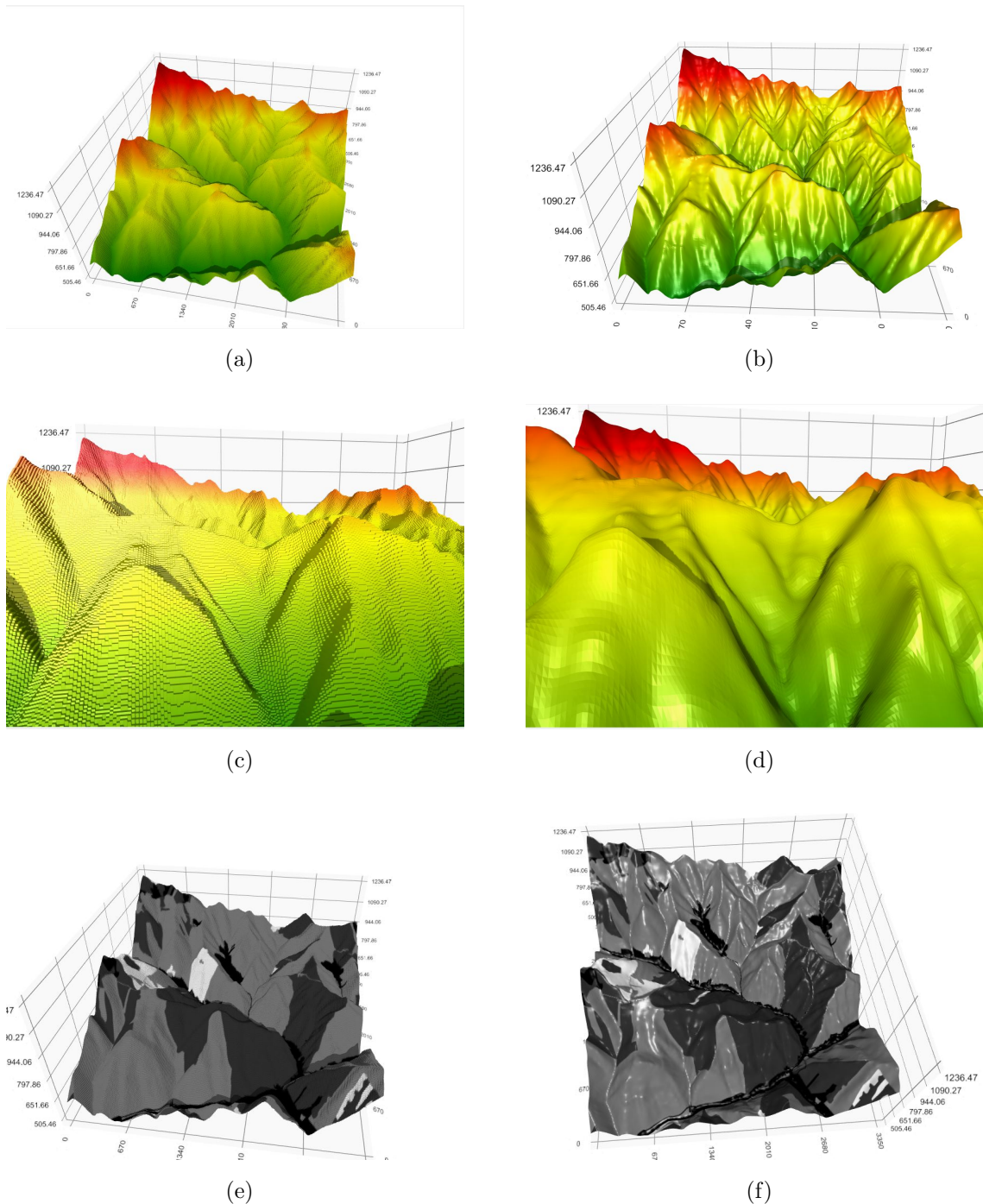
Súbory formátu VTK vieme ľahko vizualizovať v externých programoch, napríklad v programe ParaView.

4.2.4 3D vizualizácia dát

Trieda *Surface3D* obsahuje metódy a atribúty na vizualizáciu dát terénu a krivky v 3-dimenzionálnom priestore pomocou Qt knižníc *QtDataVisualisation* a *Q3DSurface*. Dáta terénnej mapy je možné zobraziť dvomi spôsobmi, vid' obr 20. Prvá možnosť je, že sa načítajú z obrázku typu *.png* po jednotlivých bitoch a tie sa pomocou príslušnej konštanty prekonvertujú na výšky v z-tovom smere. Druhá možnosť je načítavanie nadmorských výšok z textového súboru mapy terénu, ktorý sa načíta po jeho výbere. V našom konkrétnom prípade, kratší výpočtový čas zaberie druhá možnosť, pretože hodnoty nadmorských výšok sú v súbore zadávané s priestorovým krokom 10, teda do x- aj y- ového smeru ich je desaťkrát menej. Terénne dáta je možné zobraziť znovu s

4.2.4 3D vizualizácia dát

rovnakými farebnými gradientami ako v 2D prípade a navyše je možnosť zobrazenia terénu s textúrou palivovej mapy.

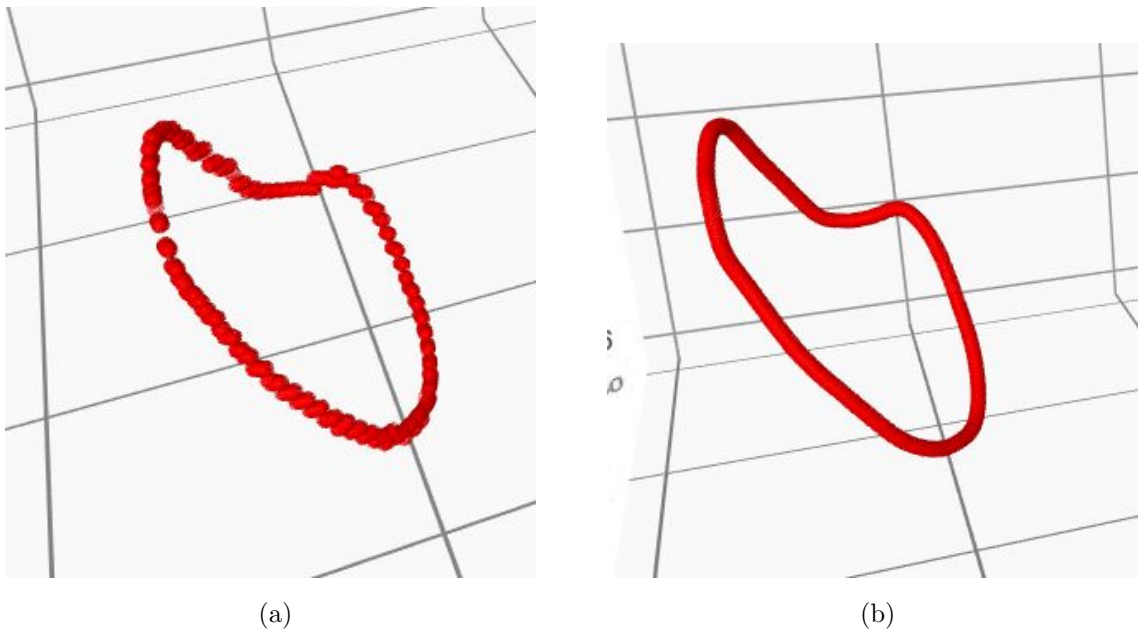


Obr. 20: Porovnanie vizualizácie terénnych dát, načítanie z obrázku (a), načítanie z textového súboru dát nadmorských výšok (b). Obrázky (c) a (d) zobrazujú detaily oboch typov zobrazení. Súradnice Na obrázkoch (e) a (f) je možné vidieť aj vizualizáciu textúry palivovej mapy na mape terénu.

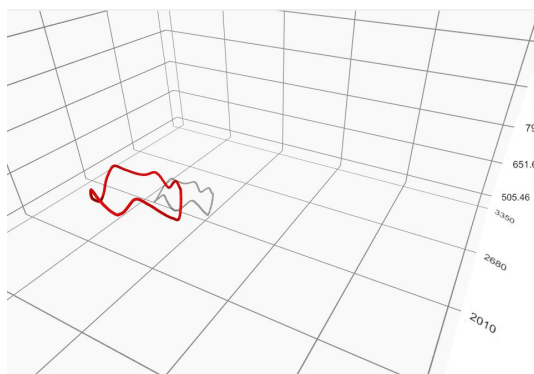
Dáta vývinu krivky idú ako vstup do konštruktoru objektu triedy. Výškové súradnice krivky sú prerátavané bilineárnou interpoláciou, kvôli jej zjemneniu vzťahom

$$\begin{aligned}
 f(x, y) \approx & \frac{f(x_1, y_1)}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) \\
 & + \frac{f(x_2, y_1)}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y_2 - y) \\
 & + \frac{f(x_1, y_2)}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y - y_1) \\
 & + \frac{f(x_2, y_2)}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y - y_1), \quad (11)
 \end{aligned}$$

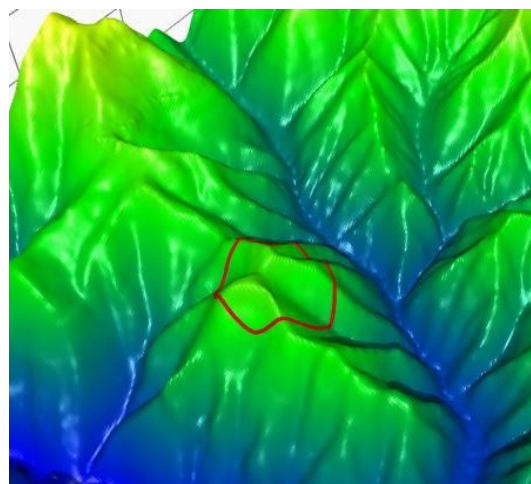
kde $(x_2 - x_1)$ a $(y_2 - y_1)$ predstavuje priestorový krok, (x, y) sú súradnice bodu krivky a $f(x_i, y_i), i = 1, 2$ sú funkčné hodnoty v uzloch okolo bodu krivky.



Obr. 21: Vľavo zobrazené body krivky bez interpolácie výškových súradníc, vpravo s interpoláciou výškových súradníc.



(a)



(b)

Obr. 22: Krivka hranice požiaru zobrazená zvlášť, šedá krivka predstavuje jej tieň (vľavo) a zobrazená na mape terénu (vpravo).

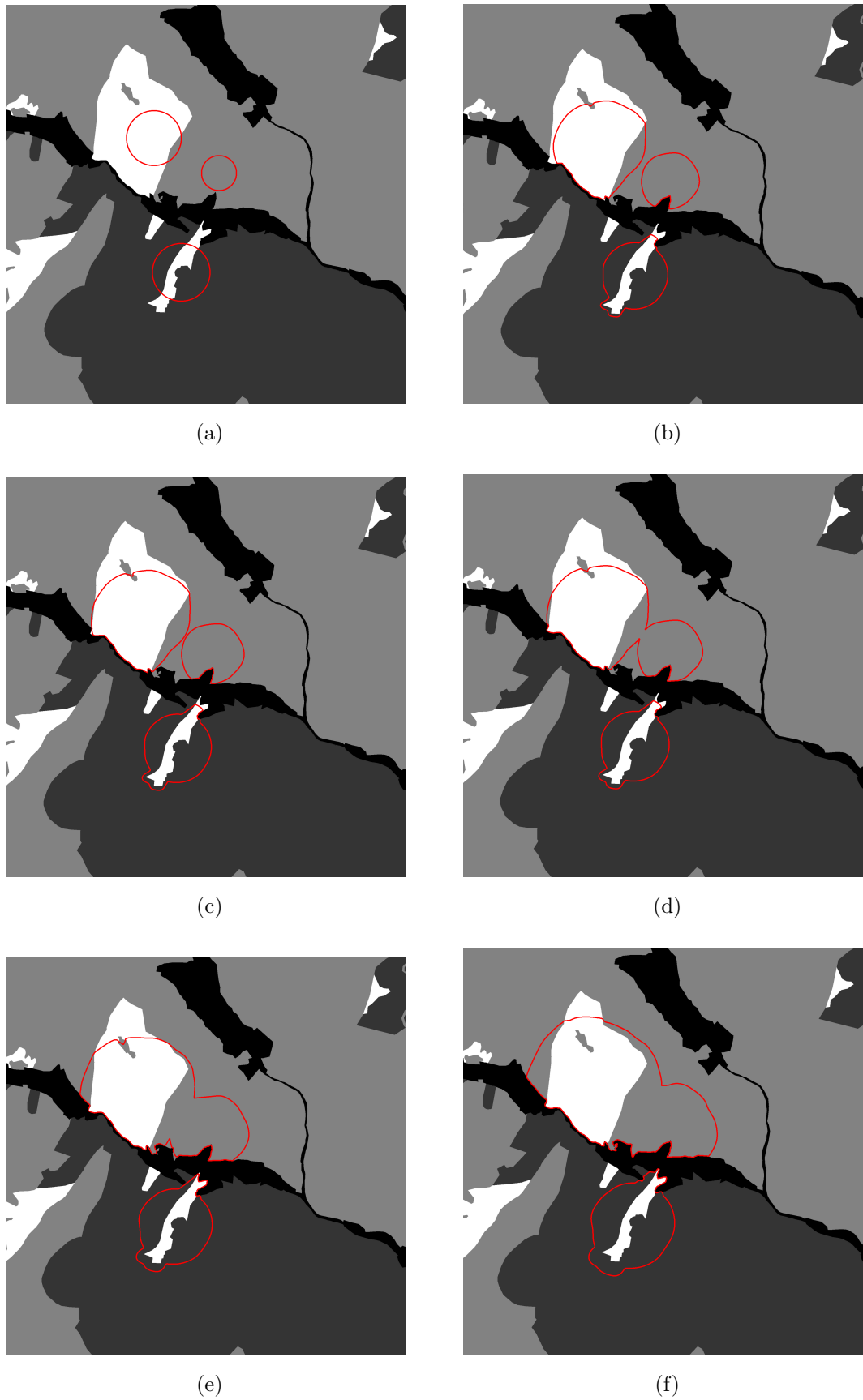
5 Praktická ukážka simulácie

Táto kapitola je zameraná na ukážku konkrétneho príkladu. Palivová a terénna mapa znázorňujú časť katastrálneho územia Staré hory v strednej časti Slovenska. Príklad bol vykonaný s počiatočnou podmienkou troch kružníc a vstupnými parametrami:

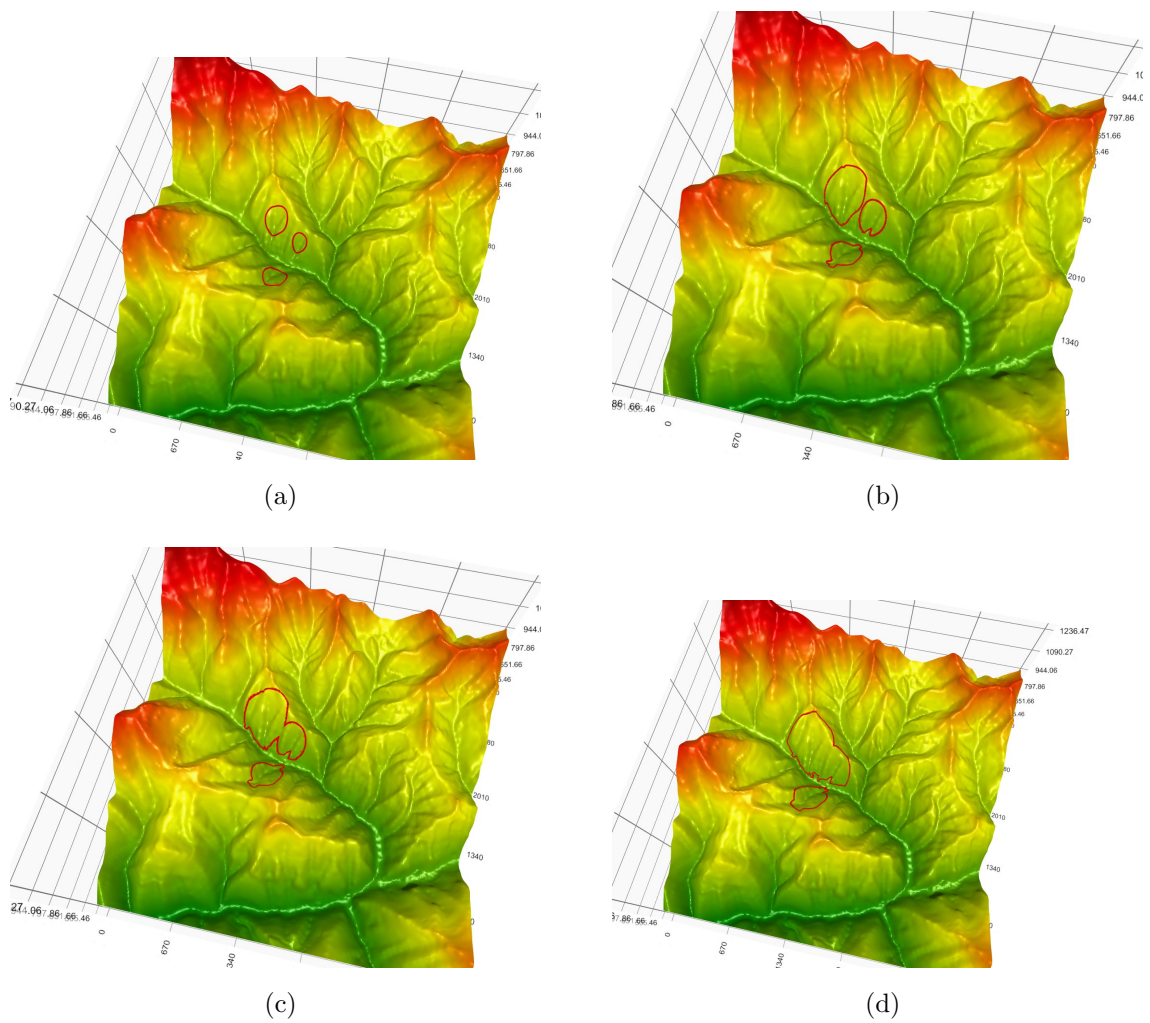
- Dĺžka časového kroku: 0.1
- Počet časových krokov: 3 000
- Maximálna normálová rýchlosť: 1
- Vplyv paliva: 1
- Vplyv geodetickej krivosti: 1
- Vplyv normálovej krivosti: 1
- Vplyv vetru: a) 0.01, b) 0.05
- Vplyv sklonu terénu: a) 0, b) 0.3
- Faktor tangenciálnej redistribúcie: 1

-
- Dĺžka jedného segmentu krivky: 0.5
 - Smer a rýchlosť vetra: severovýchodný, $60m.min^{-1}$.

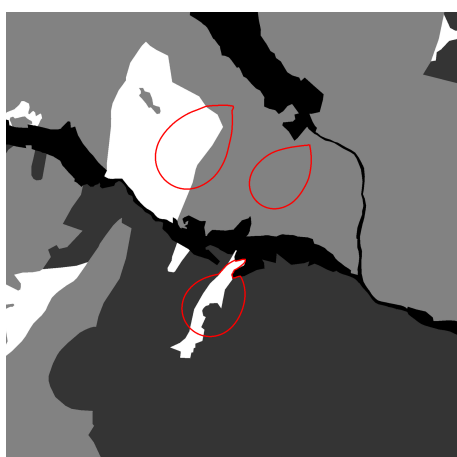
Simulácia 3000 časových krokov o dĺžke časového kroku 0.1 minúty predstavuje 5 hodín šírenia požiaru. Výpočet simulácie trval 20.34 sekúnd. Každá počiatočná kružnica je rovnomerne diskretizovaná na 800 bodov. Vizualizácie niektorých časových krokov týchto simulácií sú zobrazené na obrázkoch [23](#) až [24](#).



Obr. 23: Príklad šírenia požiaru v prípade ak je vplyv vetra 0.01 a vplyv sklonu terénu 0. Obrázky zobrazujú (a) počiatkovú podmienku a časové kroky 1000 (b), 1300 (c), 1400 (d), 2000 (e), 3000 (f).



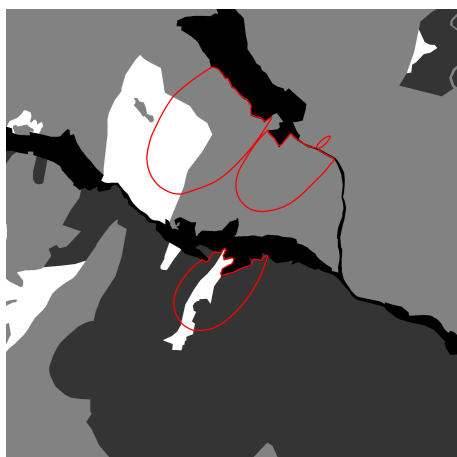
Obr. 24: Príklad šírenia požiaru vizualizovaný v 3D v prípade ak je vplyv vetra 0.01 a vplyv sklonu terénu 0. Obrázky zobrazujú (a)počiatočnú podmienku)a časové kroky 1000 (b), 1400 (c), 3000 (d).



(a)



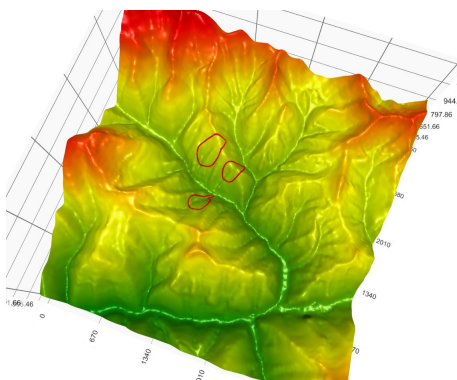
(b)



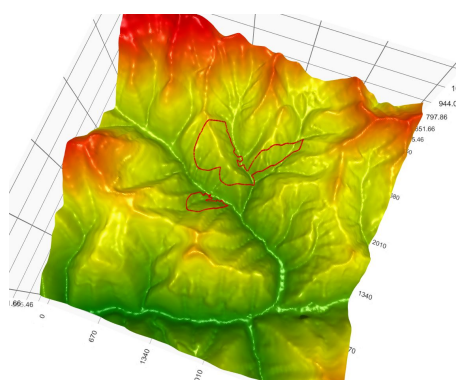
(c)



(d)



(e)



(f)

Obr. 25: Príklad šírenia požiaru v prípade ak je vplyv vetra 0.05 so severovýchodným smerom a vplyv sklonu terénu 0.3. Obrázky zobrazujú počiatočnú podmienku (a) a časové kroky 100 (b), 500 (c), 1000 (d) a 3000 (f).

6 Záver

Cieľom tejto diplomovej práce bolo naprogramovanie softvéru pre vizualizáciu modelu šírenia lesných požiarov. V úvode práce sme vypracovali prehľad rozdelenia typov iných softvérov simulujúcich šírenie požiarov a uviedli pár príkladov.

Ďalej je v práci uvedený stručný popis použitého matematického modelu šírenia lesných požiarov, ktorý bol vytvorený na Katedre matematiky a deskriptívnej geometrie Slovenskej technickej univerzity. Hranica požiaru je reprezentovaná bodmi rovnomerne diskretizovanej rovinnej krivky. Simulácia požiaru založená na tomto modeli počíta polohu krivky v nasledujúcom časovom kroku pre každý jej uzlový bod. Poloha krivky v novom časovom kroku je závislá od rôznych vplyvov prostredia, ktoré sú implementované do modelu pomocou empirických zákonov.

V jazyku C++ a s pomocou Qt knižníc sme vytvorili užívateľské prostredie, v ktorom je možné, aby si užívateľ sám volil parametre týchto vplyvov. Okrem týchto parametrov si sám zvolí počiatočné podmienky, teda ohnisko požiaru. Môže tak urobiť buď načítaním dát zo súboru formátu VTK alebo nakreslením kružnice. Výsledný časový priebeh vývoju požiaru si môže vizualizovať buď v 2D alebo 3D a ukladať. Na výber má spôsob uloženia buď ako obrázok, kde je požiar vizualizovaný na jednej z dátových máp, alebo súradnice bodov krivky vo formáte VTK. Pri vykresľovaní terénnych dát využívame aj algoritmus Marching Squares na vizualizáciu vrstevníc v 2D zobrazení.

V závere sme uviedli výstupy konkrétnych príkladov pri rôznych vstupných parametroch, čím sme poukázali na použiteľnosť softvéru na modelovanie a vizualizáciu šírenia lesných požiarov.

Do ďalšieho vývoju softvéru by sme navrhli pridať:

- možnosť vytvoriť počiatočné krivky formou polygónu,
- možnosť meniť palivovú mapu dokreslením,
- export animácie vývoja požiaru,
- export 3D súradníc kriviek do VTK súboru,
- paralelizácia programu, aby bolo možné pracovať ďalej s programom, kým by výpočet bežal na pozadí.

Zoznam literatúry

Literatúra

- [1] FINNEY, M.A.: *FARSITE: Fire Area Simulator—Model Development and Evaluation*, United States Department of Agriculture, 2004
- [2] Pacific Southwest Research Station.: *Proceedings of the Second Symposium on Social Aspects and Recreation Research*, United States Department of Agriculture, San Diego, California, 1994
- [3] MELL, W.: *Computer modelling of wildland-urban interface fires*, National Institute of Standards and Technology (NIST), San Francisco, California, 2011
- [4] MTDA team: *VESTA - Large Scale Fire Simulator*, http://www.fireparadox.org/large_scale_fire_simulator.php
- [5] AMBROZ, M.: *Numerical modelling of the forest fire propagation*, Slovak University of Technology in Bratislava Department of Mathematics and Constructive Geometry, Bratislava, 2018
- [6] Neznámy autor, *About Qt*, https://wiki.qt.io/About_Qt
- [7] WENGER, R.: *Isosurfaces: Geometry, Topology, and Algorithms* , 2013
- [8] WONG, J.: *MetaBalls and marching squares* <http://jamie-wong.com/2014/08/19/metaballs-and-marching-squares/>, 2014