|  |
| --- |
| OSTRAVSKÁ UNIVERZITA  PŘÍRODOVĚDECKÁ FAKULTA  KATEDRA INFORMATIKY A POČÍTAČŮ |
| Algoritmy kvantifikace propustnosti materiálu pro 3D rastr  DIPLOMOVÁ PRÁCE |
| Autor práce: Bc. Jan Mrógala  Vedoucí práce: Mgr. Alexej Kolcun, CSc. |
| 2022 |

|  |
| --- |
| UNIVERSITY OF OSTRAVA  FACULTY OF SCIENCE  DEPARTMENT OF INFORMATICS AND COMPUTERS |
| Material permeability quantification algorithms for 3D raster  MASTER THESIS |
| Author:  Bc. Jan Mrógala  Supervisor:  Mgr. Alexej Kolcun, CSc. |
| 2022 |

(Zadání vysokoškolské kvalifikační práce)

ABSTRAKT

Český text abstraktu

*Klíčová slova:*

*(klíčová slova vypsaná na řádku, oddělená od sebe čárkami)*

ABSTRACT

The text of the abstract.

*Keywords:*

čestné prohlášení

Já, níže podepsaný/á student/ka, tímto čestně prohlašuji, že text mnou odevzdané závěrečné práce v písemné podobě je totožný s textem závěrečné práce vloženým v databázi DIPL2.

Ostrava dne

………………………………

podpis studenta/ky

|  |
| --- |
| Poděkování |
| Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval/a samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal/a, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.  V Ostravě dne . . . . . . . . . . . .  . . . . . . . . . . . . . . . . . .  (podpis) |

OBSAH

[ÚVOD 10](#_Toc133725091)

[1 Cíle práce 11](#_Toc133725092)

[2 Definice pojmů 11](#_Toc133725093)

[2.1 Propustnost 12](#_Toc133725094)

[2.2 Úplná propustnost 12](#_Toc133725095)

[2.3 Částečná propustnost 13](#_Toc133725096)

[2.4 Kritické místo propustnosti 13](#_Toc133725097)

[3 teorie grafů 14](#_Toc133725098)

[3.1 Eulerovský graf 14](#_Toc133725099)

[3.2 Dijkstrův algoritmus 15](#_Toc133725100)

[3.3 Maximální tok a minimální řez 17](#_Toc133725101)

[3.3.1 Maximální tok 18](#_Toc133725102)

[3.3.2 Minimální řez 19](#_Toc133725103)

[4 Algoritmy hledání cesty v bludišti 20](#_Toc133725104)

[4.1 Vyplňování slepých konců (breadth-first search) 20](#_Toc133725105)

[4.2 Prohledávání do hloubky (depth-first search) 21](#_Toc133725106)

[4.3 Podobnost mezi algoritmy vyplňování a řešení bludiště 21](#_Toc133725107)

[5 Kvalita propustnosti 22](#_Toc133725108)

[5.1 Matematická morfologie 22](#_Toc133725109)

[5.1.1 Dilatace 23](#_Toc133725110)

[5.1.2 Eroze 23](#_Toc133725111)

[6 Další přístup k řešení propustnosti materiálu 24](#_Toc133725112)

[7 Otázka grafické reprezentace 25](#_Toc133725113)

[8 Algoritmus pro vyhledávání kritických míst propustnosti ve 2D 26](#_Toc133725114)

[8.1 Zamyšlení nad přístupem k problému. 26](#_Toc133725115)

[8.2 Další možné řešení problému 27](#_Toc133725116)

[9 Dopředný algoritmus pro nalezení permeability propustného materiálu 28](#_Toc133725117)

[ZÁVĚR 29](#_Toc133725118)

[RESUMÉ 30](#_Toc133725119)

[SUMMARY 31](#_Toc133725120)

[SEZNAM POUŽITÉ LITERATURY 32](#_Toc133725121)

[SEZNAM OBRÁZKŮ 34](#_Toc133725122)

[SEZNAM TABULEK 35](#_Toc133725123)

[SEZNAM PŘÍLOH 36](#_Toc133725124)

ÚVOD

Tato práce navazuje na mou bakalářskou práci, která se věnovala algoritmům pro vyplňování 3D rastru [1]. Předchozí práce se zaměřila na popis a porovnání různých vyplňovacích algoritmů, zatímco tato práce se soustředí na jejich využití při řešení konkrétního problému – permeability materiálu.

Tento problém je velmi zajímavý, protože právě pomocí vyplňovacích algoritmů můžeme detekovat permeabilitu materiálu. Konkrétněji, můžeme posoudit, zda je materiál v určitém stavu propustný, nebo nikoli. Z tohoto důvodu je tento problém ideální pro využití vyplňovacích algoritmů.

Pro praktickou aplikaci algoritmů je podobně jako v předešlé práci zvolena sada CT snímků, znázorňujících 3D pórovitý propustný materiál.

1. Cíle práce

Hlavním cílem této práce je nalezení způsobu, jak vyhodnotit permeabilitu propustného pórovitého materiálu. Důležitou roli pro vyřešení tohoto problému hrají vyplňovací algoritmy.

Vedlejším stanoveným cílem je porovnání algoritmů pro řešení bludiště a vyplňovacích algoritmů.

Pro aplikaci algoritmů je zvolen program Octave, ve kterém budou zdrojové kódy pro algoritmy napsány.

1. Definice pojmů

Pro pochopení problému permeability materiálu a jaký problém se konkrétně snažíme vyřešit si nejdříve nadefinujeme několik pojmů. Pro zjednodušení se budeme zatím bavit pouze o 2D oblasti. Pro 3D oblast se základní myšlenky těchto pojmů nemění. Nejmenším obrazovým bodem, se kterým pracujeme a bereme jej v úvahu, je pixel.

* 1. Propustnost

Chápejme propustnost materiálu v našem případě jako cestu z bodu **A** do bodu **B** přes oblast **C**. Pro ujasnění chápejme tyto body jako pixely v rastrové mřížce, zde je takto znázorňujeme pouze pro zjednodušení.

A

B

C

Obrázek 1- znázornění propustnosti materiálu

* 1. Úplná propustnost

Pro úplnou propustnost platí, že z bodu **A** existuje v dané oblasti cesta jak do bodu **B,** tak do bodu **C**.

A

B

C

Obrázek 2- znázornění úplné propustnosti

* 1. Částečná propustnost

Mějme hlavní oblast **G** obsahující podoblasti **E** a **F**, které nejsou propojeny. Částečná propustnost nastává, když z bodu **A** existuje cesta do bodu **B**, ovšem nikoli do bodu **D**.

G

A

B

C

D

E

F

Obrázek 3- znázornění částečné propustnosti

* 1. Kritické místo propustnosti

Pokud v oblasti **C** existuje místo **D**, které po uzavření znemožňuje propustnost z bodu **A** do bodu **B**, jedná se o kritické místo propustnosti v dané oblasti.

A

B

C

D

Obrázek 4- znázornění kritického místa propustnosti

I přes to, že se tato práce na obor teorie grafů nezaměřuje, je dobré si zde uvést širší souvislosti a uvědomit určité paralely, které jsou přítomny jak v problematice teorie grafů, tak hledání propustnosti v rastrovém obraze (ať už 2D či 3D).

1. teorie grafů

V teorii grafů nalézt několik algoritmů, které se zaobírají buď nalezením cesty z jednoho vrcholu do druhého (Dijkstrův algoritmus), nebo největším tokem sítí (maximální tok). Tyto problémy jsou analogické k našemu řešení problému hledání permeability v obrazovém rastru. V úvahu zde ovšem berme to, že se jedná o odlišný kontext: Práce s rastrovou mřížkou a odpovídajícími algoritmy, které pracují s pixely, oproti grafovému prostoru se svými uzly a hranami.

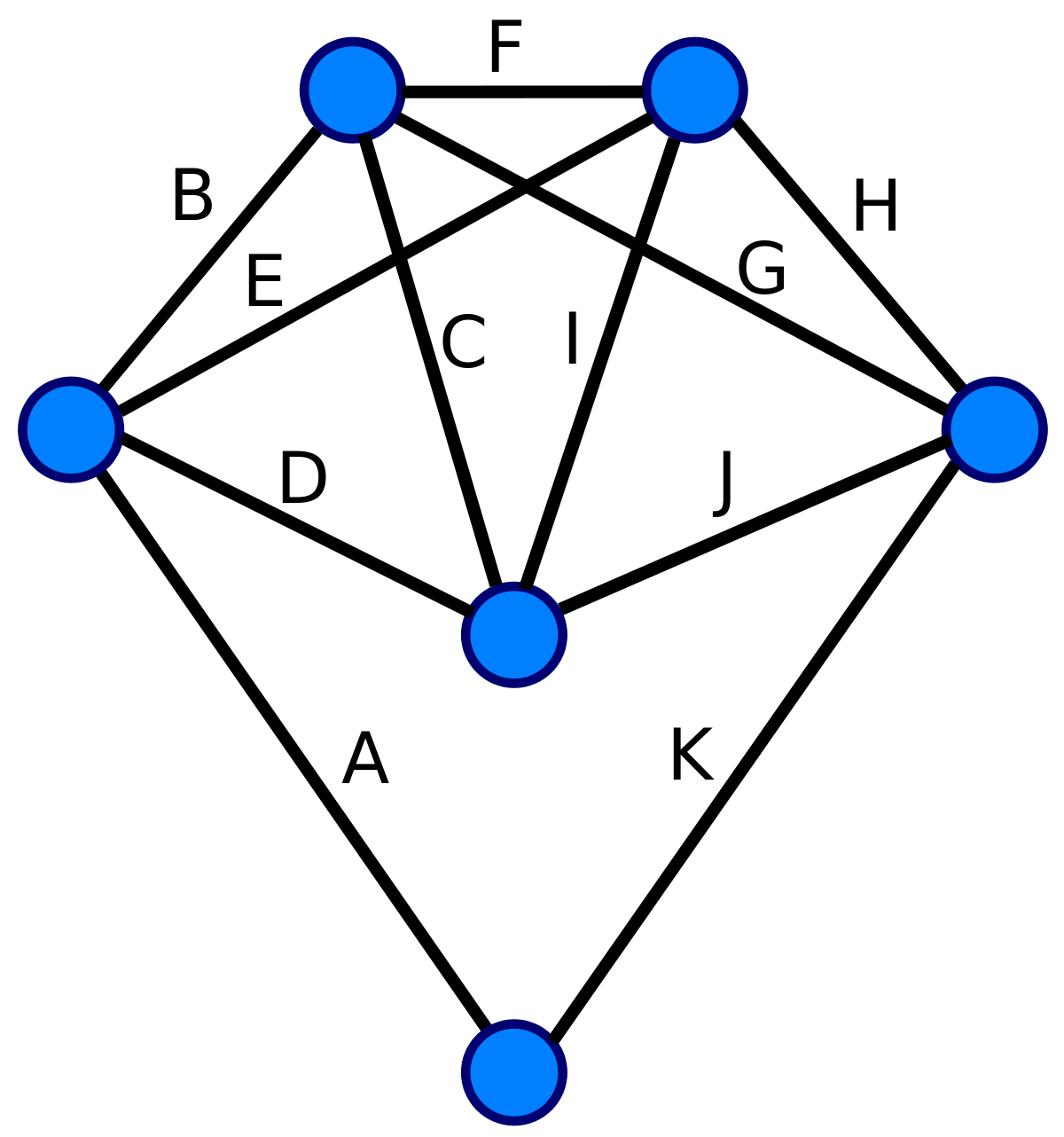
Uveďme nyní krátký úvod do teorie grafů:

* 1. Eulerovský graf

Důležitým pojmem v teorii grafů je Eulerovský graf. Ve [2] je Eulerovský graf definován následovně:

„Je to konečný graf bez izolovaných uzlů, v němž každý uzel je sudého stupně“ [2]

V Eulerově grafu existují dva typy cest, které spojují vrcholy pouze jedním průchodem v rámci grafu. Pokud cesta obsahuje každou hranu nejvýše jednou a začíná a končí ve stejném vrcholu, jedná se o uzavřenou cestu. V případě, že cesta začíná a končí v různých vrcholech, jedná se o otevřenou cestu. [2]



Obrázek 5- příklad Eulerovského grafu, převzato z

<https://cs.wikipedia.org/wiki/Eulerovsk%C3%BD_graf>

Pokud bychom se pokusili reprezentovat rastrovou mřížku jako graf, nejspíš bychom měli problém vytvořit právě graf Eulerovský. Řekněme, že bychom chápali pixel jako vrchol grafu a hrana by bylo propojení mezi nejbližšími sousedními pixely v 4-okolí. Dostali bychom se pokaždé do situace, kdy by existovaly vrcholy lichého stupně.

Jako příklad stačí uvést krajní pixel, který by měl ve svém 4-okolí pouze 3 pixely, tudíž by byl lichého stupně. Splnění podmínky pro Eulerovský graf při převodu z rastru do grafu tedy není na první pohled možný.

Reálněji se jeví možnost převést rastrovou mřížku do ohodnoceného neorientovaného grafu. Zbývá otázka, jak ohodnotit hrany. Bylo by například zajímavé vymyslet způsob hodnocení hran v kontextu rastru. Ovšem otázka, zda existuje užitečnost převodu z rastru do grafu, není v rozsahu této práce. Případně by se mohlo jednat o zajímavé téma pro práci jinou, která by tyto myšlenky mohla dále rozvíjet, zkoumat a hledat vhodné metody a využití.

* 1. Dijkstrův algoritmus

Otázku, jak se dostat z jednoho bodu do druhého do jisté míry v této práci řešíme. Více než konkrétní cesta, nás ovšem zajímá samotná existence cesty neboli propustnost. Podívejme se ale, jakým způsobem k tomuto problému přistupuje teorie grafů:

Dijkstrův algoritmus je algoritmus pro hledání nejkratší cesty v grafu s pozitivně ohodnocenými hranami. Byl vytvořen nizozemským počítačovým vědcem Edsgerem Dijkstrou v roce 1956 a publikován v roce 1959. [3]

Algoritmus funguje tak, že přiřadí dočasné vzdálenosti ke všem vrcholům v grafu a poté iterativně aktualizuje tyto vzdálenosti, dokud nenajde nejkratší cestu ze zdrojového vrcholu ke všem ostatním vrcholům. [3]

Pseudokód pro Dijkstrův algoritmus:

funkce Dijkstra(*Graf*, *zdroj, cíl*):

1. pro každý vrchol grafu v:
   1. vzdálenost[v] = NEKONEČNO
   2. předchozí[*v*] = NEDEF
   3. přidej v do Q
2. vzdálenost[*zdroj*] = 0
3. dokud Q není prázdné:
   1. u = vrchol z Q, který má nejmenší vzdálenost
   2. odstraň u z Q
   3. pokud u = cíl:
      1. S = prázdný seznam
      2. pokud předchozí[u] je DEF nebo u = zdroj:
         1. dokud u je DEF:
            1. vlož u na začátek S
            2. u = předchozí[u]
      3. ukonči cyklus
   4. pro každého souseda v od u z Q:
      1. temp = vzdálenost[u] + Graf.hrana(u, v)
      2. pokud temp < vzdálenost[v]:
         1. vzdálenost[v] = temp
         2. předchozí[v] = u
4. vrať seznam S

převzato z <https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm>

Standardně by Dijkstrův algoritmus vracel všechny vrcholy a jejich nejkratší cesty, ovšem pokud nám stačí nalézt jedinou cestu ze zdroje do cíle, můžeme hlavní cyklus ukončit předčasně, a to tak, že zkontrolujeme, zda zkoumaný vrchol odpovídá vrcholu cílovému (podmínka 3c). Následně zpětně projdeme předešlými hranami, které uložíme do seznamu S, což nám dává celou cestu ze zdroje do cíle.

* 1. Maximální tok a minimální řez

Co se naší analogie týče, v tuhle chvíli již máme způsob, jak ověřit propustnost grafu pomocí Dijkstrova algoritmu. Následně nám zbývá nalezení cesty s největším průtokem. V kontextu rastru zde chápejme propustnou cestu s nejmenším kritickým místem.

Pracujeme se sítí *S(G, s, t, c)*, která se skládá z následujících prvků:

* **G** – orientovaný graf *G = (V, E)*, skládající se z vrcholů *V* a hran *E*
* **s** – počáteční vrchol, nazýván zdroj (source)
* **t** – konečný vrchol, nazýván spotřebič (target)
* **c** – kapacita hran, *c(E)* pro konkrétní hranu daná funkcí *c: E → R+*

kromě kapacity má každý vrchol také hodnotu toku, ta se znázorňuje funkcí: *f: E → R+*. Zároveň musí platit následující 2 pravidla:



První pravidlo nám říká, že tok mezi vrcholy nemůže být záporný, nebo větší, než je kapacita hrany. Druhé pravidlo se také nazývá zákon zachování toku, protože tok v hranách ani nepřibývá, ani neubývá.

Číslo *f(e)* se nazývá tok po hraně *e*. Ve výsledku značíme hrany tímto způsobem: *x/y*, kde x je hodnota toku *f(e)* a y kapacita hrany *c(e)*. Tok přes hranu nemůže přesáhnout její kapacitu.

Řezem nazýváme množinu orientovaných hran , kde a je doplňkem.(s, t) – řezem myslíme takový řez, který odděluje zdroj od spotřebiče. [4]

A picture containing watch

Description automatically generated

Obrázek 6- příklad sítě, převzato z [4]

Věta o maximálním toku a minimálním řezu: Když existuje maximální (s, t) – tok, potom platí:

Jednoduše řečeno, maximální velikost toku v síti je dána kapacitou minimálního řezu. [4]

* + 1. Maximální tok

Maximální tok v síti představuje maximální množství "nákladu" nebo "informace", které může projít ze zdroje do cíle přes danou síť. Tento tok může být omezen kapacitou jednotlivých hran v síti. Cílem je najít největší tok, který může být přepraven z jednoho bodu do druhého.

Máme-li neorientovaný graf, nahrazujeme každou jeho hranu dvojicí orientovaných hran v opačných směrech, se stejnou hodnotou kapacity na obou.

Z věty o maximálním toku a minimálním řezu vychází Ford-Fulkersonův algoritmus, což je jeden ze základních algoritmů vylepšující cesty. Je to algoritmus pro nalezení maximálního toku a to tak, že začínáme s nulovým tokem a postupně hledáme vylepšující cesty, ve kterých tok zvětšujeme. Když už nelze tok dále zvětšit, nalezli jsme maximální tok.

Pseudokód pro Ford-Fulkersonův algoritmus:

funkce Ford-Fulkerson:

1. f = 0;
2. dokud existuje vylepšující cesta P z s do t:
   1. vylepši tok f podél cesty P
   2. f++
3. vrať f

Mějme síť *a)*, na které znázorníme průběh algoritmu. Začínáme s nulovým tokem a postupně vylepšujeme cesty. Každý obrázek ukazuje stav po navýšení toku podél červenou barvou označené cesty.

A picture containing diagram

Description automatically generated

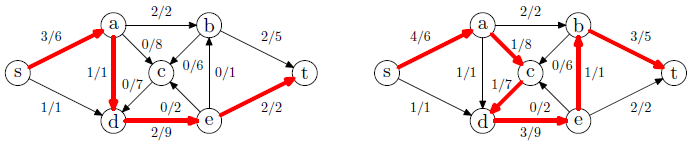
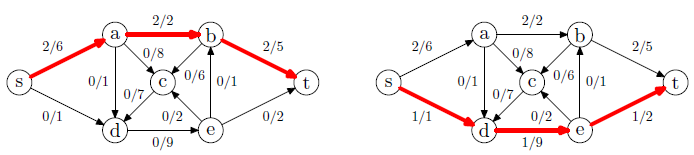
a)

d)

b)

c)

e)



Obrázek 7- Příklad postupu pro nalezení maximálního toku, převzato z [8]

Z příkladu můžeme vidět, že po čtvrtém kroku – e), již neexistuje další vylepšující cesta. Hodnota maximálního toku je v tomto případě 5.

* + 1. Minimální řez

Minimální řez je nejmenší množina hran, kterou je nutné odstranit ze sítě, aby bylo dosaženo omezení maximálního toku. Odděluje zdrojový uzel od cílového uzlu a zabraňuje toku mezi nimi. Minimální řez může být také definován jako hranová množina, která rozděluje síť na dvě disjunktní podmnožiny, z nichž jedna obsahuje zdrojový uzel a druhá obsahuje cílový uzel.

1. Algoritmy hledání cesty v bludišti

Zásadním pojmem, který jsme si definovali, bylo kritické místo propustnosti. Abychom mohli analyzovat vlastnosti tohoto místa, musíme nejdříve nalézt cestu mezi body A B. V našem případě využíváme vyplňovací algoritmy, ovšem můžeme na to taktéž nahlížet jako na problém hledání cesty bludištěm.

Za vyřešení bludiště se dá považovat nalezení cesty z počáteční buňky bludiště do cíle. Případně nás může zajímat nalezená cesta.

Základní rozdělení těchto algoritmů je podle toho, zda řešící algoritmus pracuje s celým bludištěm, nebo pouze „simuluje“ průchod bludištěm za určitých pravidel. [5]

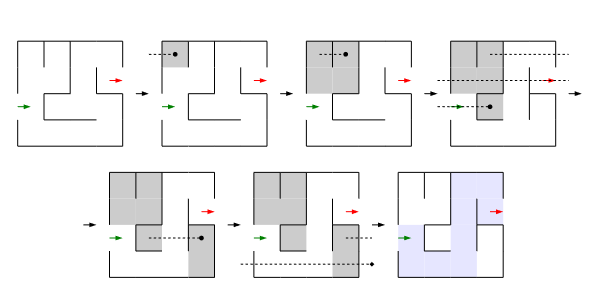
Pro lepší představu zmíníme několik existujících algoritmů, a jejich principy.

* 1. Vyplňování slepých konců (breadth-first search)

Algoritmus pracuje tak, že začne od vstupu do bludiště a postupně prochází jednotlivé chodby. Algoritmus označuje (vyplňuje) slepé cesty, které nevedou k cíli. Následně se vrací na poslední navštívenou křižovatku, ze které se na slepou cestu vydal. Poté pokračuje po jiné nevyzkoušené cestě a opakuje tento postup, dokud nenajde cestu k východu z bludiště. [5]

Jedná se o jednoduchý algoritmus, za jehož největší nevýhodu můžeme pokládat nutnost prohledat celé bludiště.

U tohoto algoritmu můžeme pozorovat podobnost s řádkovým semínkovým vyplňováním právě ve využití zapamatování křižovatky (kritického pixelu), ke kterému se algoritmus vrací.



Obrázek 8- průběh algoritmu slepých konců, převzato z [5]

* 1. Prohledávání do hloubky (depth-first search)

Prohledávání do hloubky prohledává graf postupně podle jednotlivých větví, tedy prochází do hloubky jednu větev, dokud nenarazí na uzel bez nezpracovaných sousedů. Poté se vrací zpět k poslednímu uzlu, u kterého existuje nezpracovaný soused a opakuje tento proces znovu. [5]

Prohledávání do hloubky nemusí najít nejkratší cestu k cílovému uzlu, ale může najít řešení rychleji než vyplňování slepých konců, pokud je cílový uzel hlouběji v grafu. [5]

Diagram

Description automatically generated

Obrázek 9- Postup prohledávání do hloubky, převzato z

<https://www.simplilearn.com/tutorials/data-structure-tutorial/dfs-algorithm>

* 1. Podobnost mezi algoritmy vyplňování a řešení bludiště

Oba algoritmy, řešení bludiště i výplně vybrané oblasti, zahrnují průchod dvourozměrným prostorem a prozkoumávání sousedních buněk nebo pixelů. Oba také zahrnují použití datových struktur, jako jsou zásobníky nebo fronty, pro udržování navštívených buněk nebo pixelů a pro řízení procesu prohledávání nebo výplně.

Existují však také některé klíčové rozdíly mezi algoritmy řešení bludiště a výplně vybrané oblasti. Algoritmy řešení bludiště se obvykle zaměřují na nalezení jediné cesty z počátečního bodu do koncového bodu, zatímco algoritmy výplně vybrané oblasti se zaměřují na vyplnění celé oblasti zájmu.

1. Kvalita propustnosti

Pod otázkou kvality propustnosti permeabilního materiálu si představíme především výskyt kritických míst, a jak silná je propustnost, respektive velikost kritického místa. Pro vyřešení těchto problémů využijeme matematickou morfologii.

* 1. Matematická morfologie

Matematická morfologie je matematická teorie, která se zabývá analýzou geometrických transformací, jako jsou eroze, dilatace, otevření a uzavření, aplikovaných na obrazy nebo jiné geometrické objekty. Tato teorie byla poprvé navržena Georgesem Matheronem v roce 1962 a později rozvinuta Jeanem Serra v 70. letech 20. století. [6]

Základní myšlenkou matematické morfologie je aplikace geometrických transformací na obrazová data za účelem získání informací o struktuře, tvaru a topologii objektů v obraze. Eroze a dilatace jsou základními operacemi matematické morfologie, které slouží k úpravě tvaru objektů v obraze. [6]

Morfologické operace vychází z myšlenky, že binární obraz můžeme vyjádřit jako množinu 2D bodů. Této množině říkejme X. Každý bod v této množině reprezentuje pixel v obraze s hodnotou jedna. Jako příklad uveďme následující množinu X = {(1, 1), (1, 2), (2, 1),  
(2, 2)}.

Transformace obrazu je relace mezi dvěma takovými množinami bodů, kde ta druhá je obvykle menší a nazývá se strukturní element. Transformace můžeme provádět jak na 2D tak i 3D obraze.

A picture containing shape

Description automatically generated

Obrázek 10 - příklad strukturních elementů, převzato z [6]

* + 1. Dilatace

Pro tuto práci je dilatace nejužitečnější operace matematické morfologie. Za pomocí vektorového součtu mezi množinami X a B expandujeme objekty v obraze. Předpis dilatace je následující:



Obrázek 11- Příklad dilatace, převzato z [6]

* + 1. Eroze

Eroze je duální operace k dilataci. Nejedná se ovšem o invertovanou dilataci, obě tyto operace nejsou invertovatelné. Předpis eroze je následující:

B = {}

A picture containing text, clipart

Description automatically generated

Obrázek 12- Příklad eroze, převzato z [6]

1. Další přístup k řešení propustnosti materiálu

Problém propustnosti materiálu je ve vědecké komunitě většinou řešen odlišnými způsoby. Konkrétně je nejčastější laboratorní, či fyzikální přístup. Případně modelování chování proudění kapaliny v propustném materiále.

Jedna z prací, kterou můžeme na toto téma nalézt, kombinuje laboratorní přístup s modelováním.

Uveďme si zde, o čem se [7] zabývá:

Autoři kombinovali pokročilé techniky charakterizace a modelování k analýze mikrostruktury otevřeně porézních páskově litých materiálů a jejich vlivu na propustnost. Byly vyrobeny čtyři vzorky s různým obsahem porogenu pomocí metody páskového lití. Následné vypalování vyrobených zelených pásek umožnilo získat otevřeně porézní struktury s porozitou v rozmezí 45–50%. [7]

Byla provedena kvantitativní analýza obrazu 3D mikro-počítačové tomografie a byly získány podrobné charakteristiky porézní mikrostruktury. Na základě získaných dat byl vyvinut reprezentativní model otevřeně porézní mikrostruktury. Metoda konečných objemů byla použita k výpočtu propustnosti různých scénářů mikrostruktury. [7]

Získané výsledky ukazují, že porozita, průměrná velikost pórů a konstriktivita přímo ovlivňují vlastnosti proudění kapalin. Významné změny v konstriktivitě a výsledné propustnosti mohou být způsobeny významným přídavkem porogenu, pro který se mění propojení pórů a tvorba (nebo její absence) volných cest ovlivňuje proudění kapalin. [7]

1. Otázka grafické reprezentace

Jelikož tato práce manipuluje s obrazovými daty, bylo by vhodné prozkoumat možnosti grafické reprezentace výstupu z algoritmů. Inspirovat se můžeme prací [8].

1. Algoritmus pro vyhledávání kritických míst propustnosti ve 2D
   1. Zamyšlení nad přístupem k problému.

Způsoby, jak vyhledat kritická místa propustnosti můžeme vymyslet více. Nejjednodušší algoritmus, který by se dal použít, by mohl při vyplňování zároveň prohledávat okolní čtyři sousední pixely zrovna vyplňovaného pixelového bodu. Narážíme zde ovšem na více problémů. Ten první je, že by nestačilo prosté kontrolování sousedního okolí pixelu, ale bylo by nutné prohledávat větší oblast, jelikož nemůžeme předpokládat, že v materiálu je kritické místo propustnosti o velikosti 1 pixelu. Další problematická situace nastává při určité konfiguraci hraniční oblasti.

Diagram

Description automatically generated with medium confidence

Obrázek 13: 4 směry hledání

Problémový případ:

Chart, waterfall chart

Description automatically generated

Obrázek 14: problém 4 směrů hledání

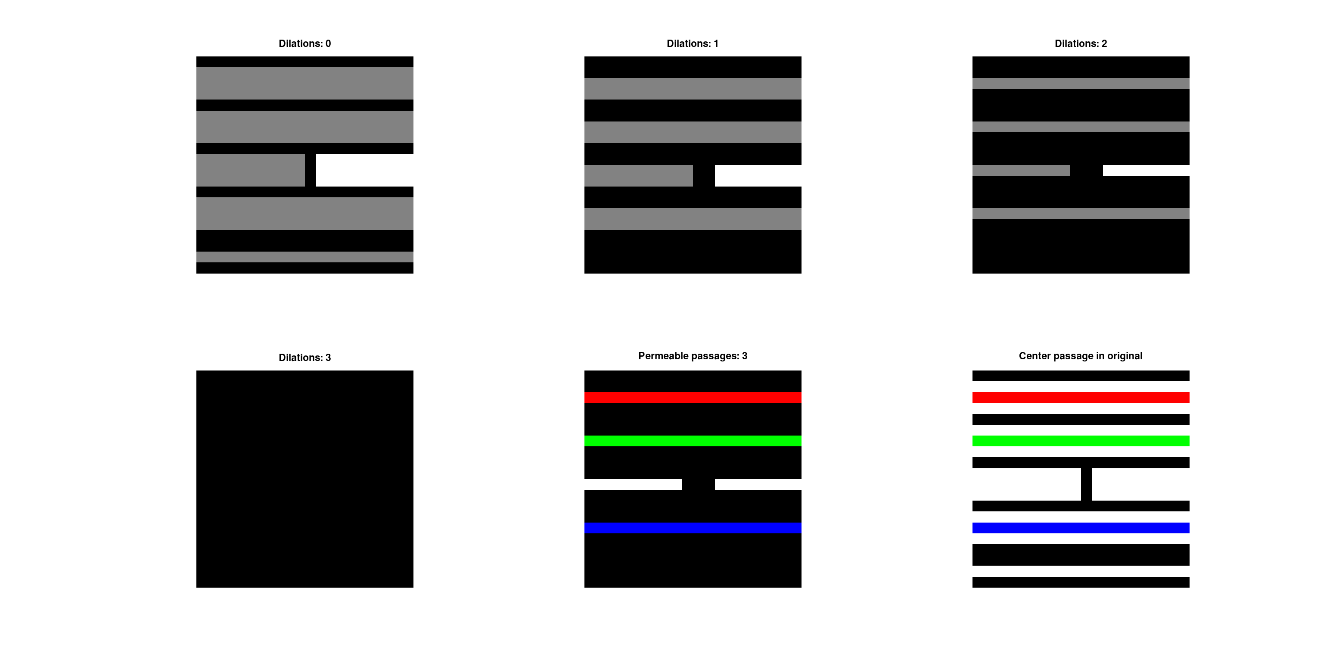
Velký problém, na který narážíme u jednoduchého prohledávání sousedních pixelů je ten, že existují případy, kdy je kritické místo těžko detekovatelné. Příklad můžeme vidět na obrázku výše. Pokud bychom vyhledávali pouze do 4 směru, tak nedetekujeme kritické místo mezi dvěma obdélníky. Tento případ je tedy důvodem, proč využití této jednoduché metody není dobrý nápad. Další způsoby, jak prohledávat okolí jednoho bodu, jistě existují, ovšem komplexnost začíná růst jak z implementačního hlediska, tak výpočetního. Přesuneme se tedy na další možnost řešení daného problému.

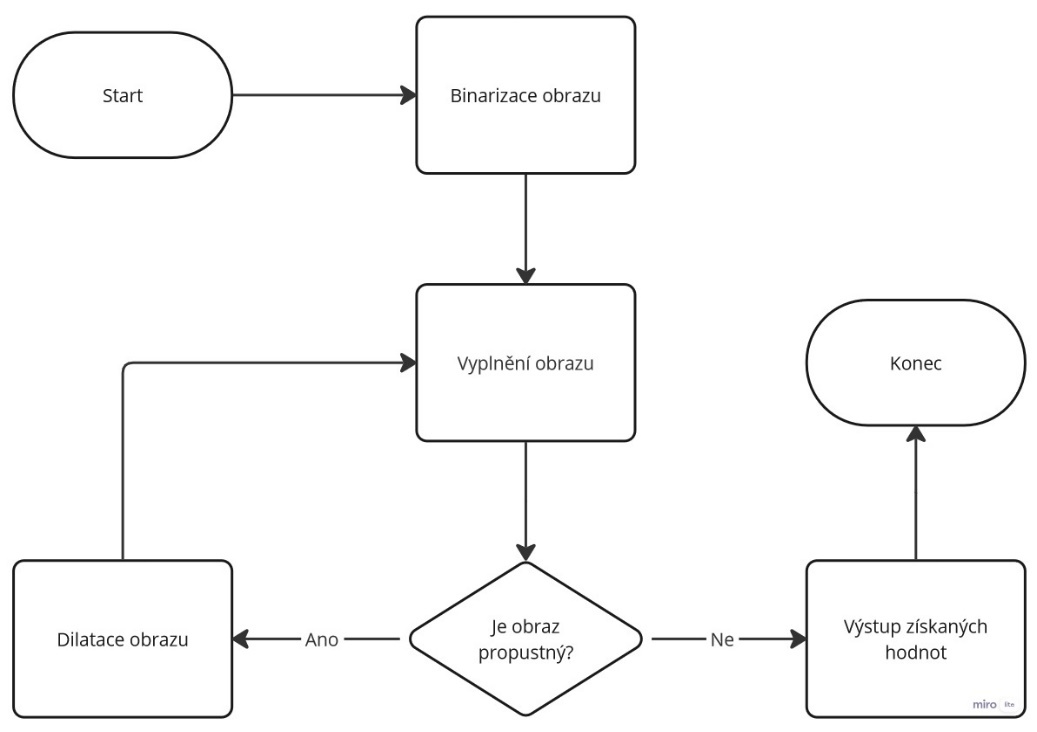
* 1. Další možné řešení problému

Pokud nejsme reálně schopni využít prohledávání sousedního okolí kolem pixelového bodu, což by byl nejjednodušší přístup, vymyslíme jiný.

Následující přístup využívá teorie matematické morfologie, a to konkrétně dilatace. Myšlenka spočívá v tom, že místo toho, abychom zkoumali jednotlivé pixely, zda se nejedná o kritické místa, raději postupně dilatujeme hraniční body. Tento přístup je méně výhodný v tom, že bude nutné opakované vyplňování zkoumané oblasti, abychom zjistili, zda došlo k uzavření materiálu. Tato operace rozšiřování bude opakována tak dlouho, až dojde k uzavření. Jakmile zjistíme, že již nelze materiálem prostoupit na jeho druhý konec, víme, že jsme zaplnili místo, které je kritické.

1. Dopředný algoritmus pro nalezení permeability propustného materiálu





ZÁVĚR

RESUMÉ

SUMMARY

SEZNAM POUŽITÉ LITERATURY

1. **Mrógala J.:** *Algoritmy vyplňování pro 3D rastr*. Ostrava, 2021. Bakalářská práce. Ostravská univerzita. Přírodovědecká fakulta. Katedra informatiky a počítačů.
2. **Sedláček J.:** *Úvod do teorie grafů*. Academia, Praha, 1977. ISBN 510-21-826.
3. **Bělohlávek R., Vychodil V.:** *Diskrétní matematika pro informatiky II.* Olomouc: PřF UP Olomouc, 2006.
4. **Černý J.:** *Základní grafové algoritmy*. Praha, 2010. KAM, MFF UK.
5. **Matějka P.:** *Algoritmy pro generování a řešení bludišť*. Brno, 2012. Diplomová práce. Masarykova univerzita. Fakulta informatiky.
6. **Šonka M., Hlaváč V., Boyle R.:** *Image Processing, Analysis and Machine Vision*. Springer 2015.
7. **Haj Ibrahim, S., Skibinski J., Oliver G. J., Wejrzanowski T.:** *Microstructure effect on the permeability of the tape-cast open-porous materials*. Materials & Design Vol. 167, April 2019.
8. **Grossmanová P.:** *Model 3D prostoru*. Ostrava, 2019. Diplomová práce. Ostravská univerzita. Přírodovědecká fakulta. Katedra informatiky a počítačů.
9. SEZNAM POUŽITÝCH SYMBOLŮ

|  |  |  |  |
| --- | --- | --- | --- |
| ABC |  |  | Význam první zkratky. |
| B |  |  | Význam druhé zkratky. |
| C |  |  | Význam třetí zkratky. |
|  |  |  |  |

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH