|  |
| --- |
| 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](#_Toc132777767)

[1 Cíle práce 11](#_Toc132777768)

[2 Definice pojmů 11](#_Toc132777769)

[2.1 Propustnost 12](#_Toc132777770)

[2.2 Úplná propustnost 12](#_Toc132777771)

[2.3 Částečná propustnost 13](#_Toc132777772)

[2.4 Kritické místo propustnosti 13](#_Toc132777773)

[3 teorie grafů 14](#_Toc132777774)

[3.1 Eulerovský graf 14](#_Toc132777775)

[3.2 Dijkstrův algoritmus 15](#_Toc132777776)

[3.3 Maximální tok a minimální řez 17](#_Toc132777777)

[3.3.1 Maximální tok 17](#_Toc132777778)

[3.3.2 minimální řez 17](#_Toc132777779)

[4 Algoritmy hledání cesty v bludišti 18](#_Toc132777780)

[4.1 Vyplňování slepých konců (breadth-first search) 18](#_Toc132777781)

[4.2 Prohledávání do hloubky (depth-first search) 19](#_Toc132777782)

[5 Kvalita propustnosti 20](#_Toc132777783)

[5.1 Matematická morfologie 20](#_Toc132777784)

[5.1.1 Dilatace 20](#_Toc132777785)

[5.1.2 Eroze 20](#_Toc132777786)

[6 Podobnost mezi algoritmy vyplňování a řešení bludiště 21](#_Toc132777787)

[7 Jiné metody a přístupy k řešení propustnosti materiálu (Rešerše) 21](#_Toc132777788)

[8 Otázka grafické reprezentace 21](#_Toc132777789)

[9 Algoritmus pro vyhledávání kritických míst propustnosti ve 2D 22](#_Toc132777790)

[9.1 Zamyšlení nad přístupem k problému. 22](#_Toc132777791)

[9.2 Další možné řešení problému 23](#_Toc132777792)

[ZÁVĚR 24](#_Toc132777793)

[RESUMÉ 25](#_Toc132777794)

[SUMMARY 26](#_Toc132777795)

[SEZNAM POUŽITÉ LITERATURY 27](#_Toc132777796)

[SEZNAM OBRÁZKŮ 29](#_Toc132777797)

[SEZNAM TABULEK 30](#_Toc132777798)

[SEZNAM PŘÍLOH 31](#_Toc132777799)

ÚVOD

Tato práce navazuje na mou bakalářskou práci, která se věnovala algoritmům pro vyplňování 3D rastru [4]. 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

V jinem kontextu zminit

1. teorie grafů

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).

Konkrétně můžeme 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 – **rozepsat rastr vs graf – v úvahu říct až po grafech**

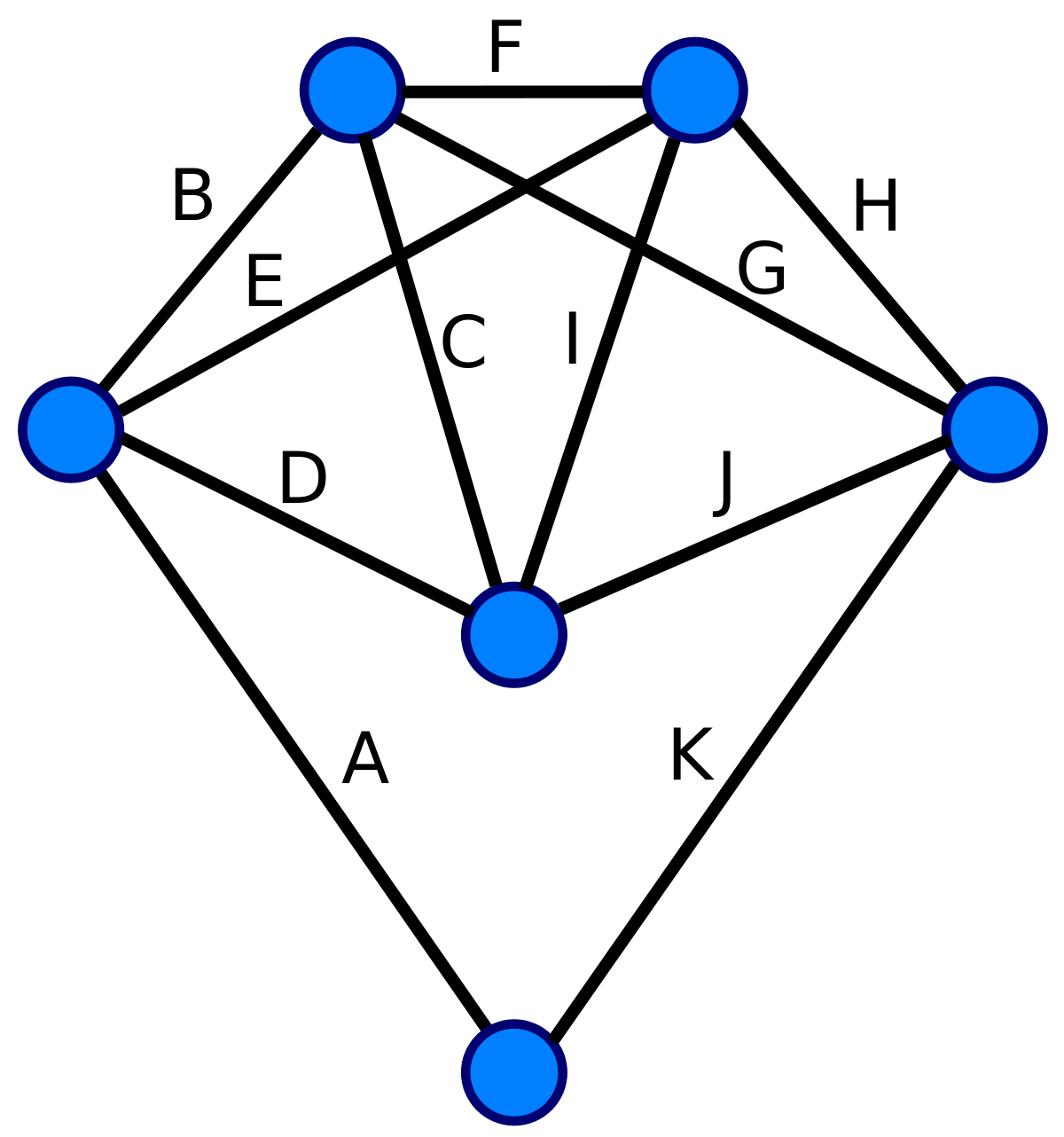
Uveďme nyní krátký úvod do teorie grafů:

* 1. Eulerovský graf

Důležitým pojmem v teorii grafů je Eulerovský graf. v [6] definuje eulerovský graf následovně:

„Je to konečný graf bez izolovaných uzlů, v němž každý uzel je sudého stupně“ [6]

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. [6]



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. [7]

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. [7]

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.

Co se naší analogie týče, v tuhle chvíli již máme způsob, jak ověřit propustnost grafu. Následně nám zbývá nalezení maximálního toku grafem. K tomu si přiblížíme techniku maximálního toku a minimálního řezu.

* 1. Maximální tok a minimální řez

Maximální tok a minimální řez jsou dva základní koncepty v teorii grafů a teorii sítí.

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.

Minimální řez je pak naopak nejmenší množina hran, kterou je nutné odstranit ze sítě, aby bylo dosaženo omezení maximálního toku. Jinými slovy, je to nejmenší množina hran, která odděluje zdrojový uzel od cílového uzlu a zabraňuje toku nákladu 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. Maximální tok
    2. Minimální řez

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]

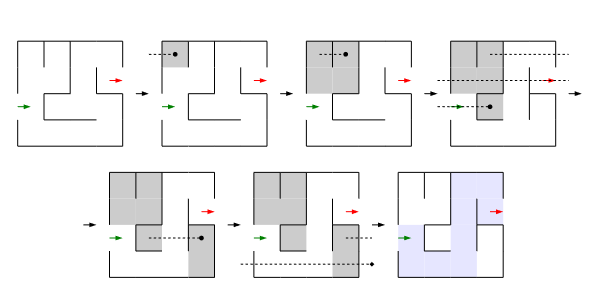
Pro lepší představu zmíníme několik existujících algoritmů, a jejich principy.

* 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.

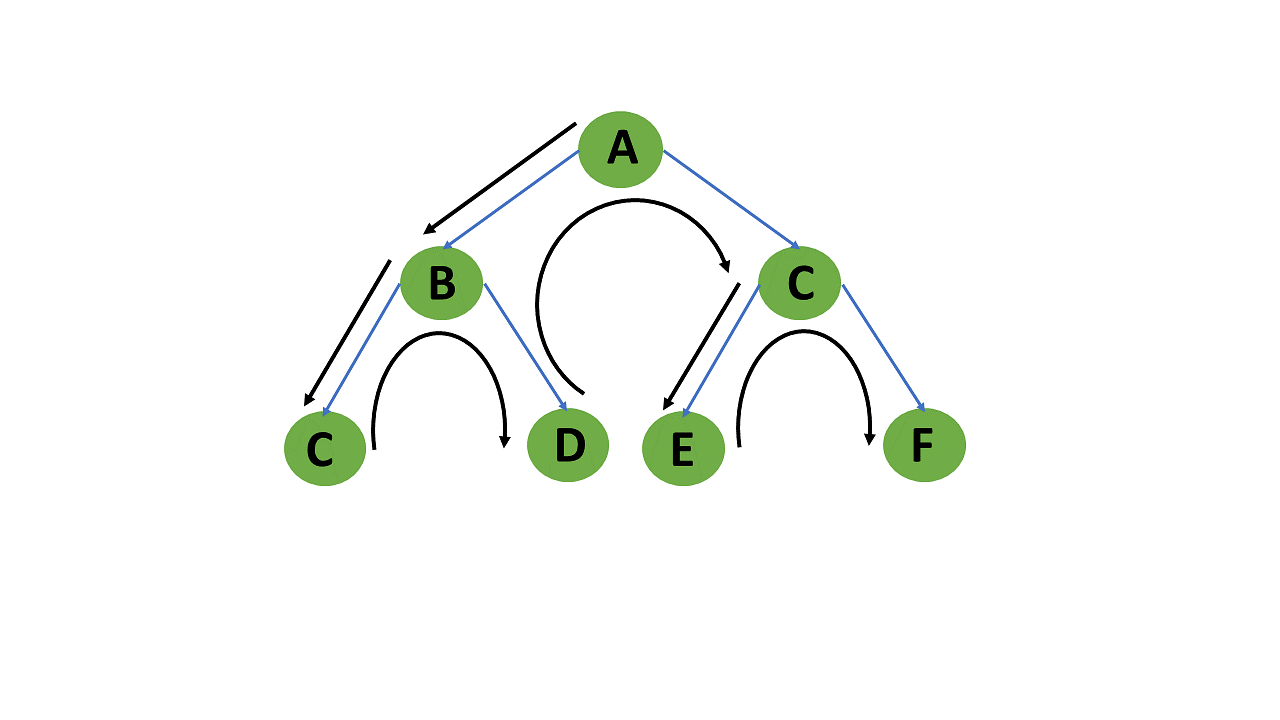


Obrázek 6- průběh algoritmu slepých konců, převzato z [5]

* 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.

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.



Obrázek 7- Postup prohledávání do hloubky, převzato z

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

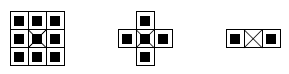
1. Kvalita propustnosti

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. [2]

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.



Obrázek 8 - příklad strukturních elementů [2]

* + 1. Dilatace
    2. Eroze

Zmínit o matlabu dilatace, kapitolu 6 dát před 5

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. Jiné metody a přístupy k řešení propustnosti materiálu (Rešerše)

Problém propustnosti materiálu je ve vědecké komunitě většinou řešen odlišnými způsoby než tato práce. 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 tato práce s názvem „*Microstructure effect on the permeability of the tape-cast open-porous materials”* zabývá:

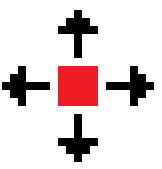
Autoři kombinovali pokročilé techniky charakterizace a modelování k analýze mikrostruktury otevřeně porézních páskově litých materiálů a jejich vlivu na propustnost. Byly vyrobeny čtyři vzorky s různým obsahem porogenu pomocí metody páskového lití. Následné vypalování vyrobených zelených pásek umožnilo získat otevřeně porézní struktury s porozitou v rozmezí 45–50%. [1]

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. [1]

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. [1]

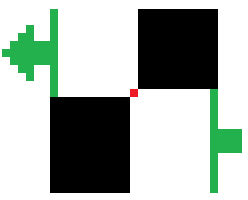
1. Otázka grafické reprezentace
2. Algoritmus pro vyhledávání kritických míst propustnosti ve 2D
   1. Zamyšlení nad přístupem k problému.

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.



Obrázek 9: 4 směry hledání

Problémový případ:



Obrázek 11: 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.

ZÁVĚR

RESUMÉ

SUMMARY

SEZNAM POUŽITÉ LITERATURY

Seřadit podle přijmení a změnit pořadí příjmení jméno

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

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

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH