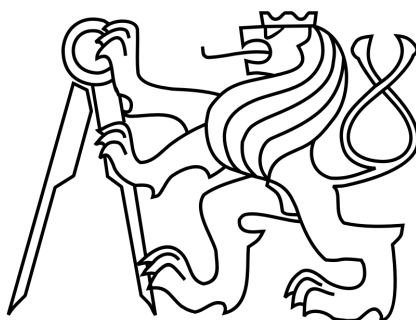


České vysoké učení technické v Praze

Fakulta stavební

Katedra mechaniky



BAKALÁŘSKÁ PRÁCE

Geometrické modelování heterogenních
materiálů založené na Wangových dlážděních

Vedoucí práce: Doc. Ing. Jan Zeman, Ph.D.

Praha 2012

Martin Doškář

Název práce: Geometrické modelování heterogenních materiálů založené na Wangových dlážděních
Autor: Martin Doškář
Katedra: Katedra mechaniky
Vedoucí práce: Doc. Ing. Jan Zeman, Ph.D.

Abstrakt: V práci je představen koncept Wangova dláždění, který umožňuje vytvářet přirozené, náhodné textury bez viditelných artefaktů, a to z poměrně malého souboru dlaždic. Práce zkoumá jeho využití pro kompresi a s ní spojenou rekonstrukci mikrostruktury heterogenních materiálů. Dále je prezentován Image Quilting algoritmus a jeho aplikace pro automatickou konstrukci Wangových dlaždic. Je navržena úprava tohoto algoritmu pro vícefázová média s analytickými tvary inkluzí. Za účelem porovnání referenčních a rekonstruovaných mikrostruktur jsou v práci představeny základy teorie a způsobu výpočtu statistických deskriptorů, zejména pak dvoubodové pravděpodobnostní funkce.

Klíčová slova: Wangovo dláždění, komprese a rekonstrukce mikrostruktury, modelování heterogenních materiálů, Image quilting algoritmus

Title: Geometrical modelling of hererogenous materials based
on Wang tilings
Author: Martin Doškář
Department: Department of Mechanics
Supervisor: Doc. Ing. Jan Zeman, Ph.D.

Abstract: This thesis presents the concept of Wang Tiling, capable of creating naturally looking patterns from a small set of tiles. It discusses its applicability to compression and reconstruction of heteregenous material microstructures. To this end, Image Quilting algorithm is implemented for automatic design of Wang Tiles. Modifications of stochastic tiling algorithm and Image Quilting algorithm are proposed and discussed from the viewpoint of the quality of reconstructed media. To facilitate a comparison between reference and reconstructed media, basic theory and evaluation techniques of spatial statistics are introduced. Particular focus is on the two-point probability function.

Keywords: Wang tilings, microstructure compression and reconstruction, heterogenous materials, Image quilting algorithm

Prohlašuji, že jsem tuto bakalářskou práci napsal samostatně, pouze za odborného vedení vedoucího práce doc. Ing. Jana Zemana, Ph.D. Dále prohlašuji, že veškeré podklady, ze kterých jsem čerpal, jsou uvedeny v seznamu literatury.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 9. 5. 2012

Martin Doškář

Rád bych poděkoval vedoucímu bakalářské práce doc. Ing. Janu Zemanovi, Ph.D., za rady a čas, který mi věnoval při přípravě nejen této bakalářské práce. Dále bych chtěl poděkovat své rodině za podporu.

Tato práce byla podpořena projektem číslo P105/11/0411, uděleným Grantovou agenturou České republiky.

Obsah

Použité symboly a názvosloví	vi
Úvod	1
1 Statistický popis mikrostruktury	4
1.1 Dvoubodová pravděpodobnostní funkce	6
1.2 Metody výpočtu dvoubodové pravděpodobnosti	7
1.3 Shrnutí	11
2 Komprimovaná reprezentace mikrostruktury	12
2.1 Komprimace založená na Wangových dlaždicích	12
2.2 Stochastické dláždění	13
2.3 Shrnutí	17
3 Konstrukce Wangových dlaždic	19
3.1 Konstrukce dlaždic pomocí algoritmu Image Quilting	19
3.2 Shrnutí	25
Závěr	26
Literatura	30
Seznam obrázků	32
Seznam algoritmů	33
A Implementace v prostředí Matlab	34
B Algoritmus otáčení a ořezávání dlaždic	48
C Diskrétní Fourierova transformace	51

Použité symboly a názvosloví

α	Jedna realizace mikrostruktury ze souboru mikrografů S
$\overline{F(\mathbf{x})}$	Průměr funkce přes celý soubor S
$\widetilde{F(\mathbf{x})}$	Průměr funkce přes celou oblast V jediné realizace α
$\chi_i(\mathbf{x})$	Charakteristická funkce i -té fáze
ϕ_i	Objemové zastoupení i -té fáze
$f^*(\mathbf{x})$	Funkce komplexně sdružená k funkci $f(\mathbf{x})$
$p(\alpha)$	Pravděpodobnost výskytu realizace α v souboru mikrografů S
S	Soubor mikrografů (Ensemble)
$S_i(\mathbf{x})$	Jednobodová pravděpodobnostní funkce
$S_{rs}(\mathbf{x}_1, \mathbf{x}_2)$	Dvoubodová pravděpodobnostní funkce
DFT	Diskrétní Fourierova transformace (Discrete Fourier Transform)
FFT	Fast Fourier Transform
FT	Fourierova transformace (Fourier Transform)
RVE	Reprezentativní objemový vzorek (Representative Volume Element)
SVE	Statistický objemový vzorek (Statistical Volume Element)

Jelikož jsou veškeré dostupné podklady věnující se této tématice psány v anglickém jazyce, bylo nutné zavést překlady některých termínů. Koncept vytváření roviny z navazujících buněk je v této práci nazýván dlážděním (koresponduje s anglickým termínem tiling). Základní buňky jsou nazývány dlaždice, případně Wangovy dlaždice (v originále Wang Tiles). Metoda spojení dvou obrázků představená v [1] a použitá k automatické tvorbě dlaždic

zůstala nepřeložena a je uváděna jako Image Quilting. Některé statistické deskriptory uvedené v kapitole 1 zůstávají také nepřeloženy. U deskriptorů s přeloženým názvem zůstává původní anglický název uveden v závorkách.

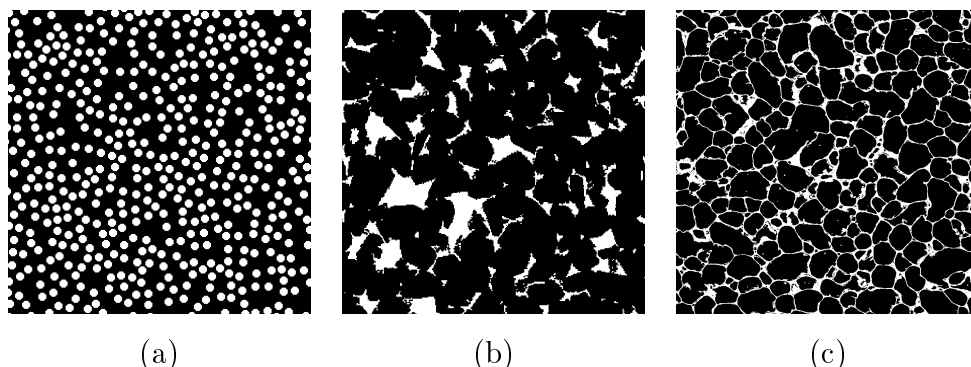
V práci se vyskytují čtyři úrovně grafické reprezentace mikrostruktury materiálu. Referenční médium je základní záznam získaný buď skenováním, či magnetickou resonancí. Z tohoto média jsou vybrány reprezentativní vzorky, které slouží jako vstupní data pro automatickou tvorbu dlaždic. Wangovy dlaždice jsou další úrovní, přičemž jejich velikost závisí na velikosti reprezentativního vzorku a velikosti přesahu použitého při spojování vzorků. Posledním typem je rekonstruované médium, které vzniká dlážděním roviny.

Úvod

Rozvoj výpočetní techniky a hlubší poznání v oblasti chování materiálů umožnily návrh konstrukcí, které by dříve byly stěží proveditelné. S tímto rozvojem úzce souvisí studium nových materiálů se specifickými termomechanickými vlastnostmi, jako jsou například vysoká mez pevnosti či kluzu, modul přetvárnosti, nízká hmotnost, tepelně-izolační vlastnosti, a pokud možno nízké výrobní náklady [2]. Značný potenciál v tomto směru vykazují kompozitní materiály, které při správném návrhu dokáží kombinovat a optimálně využít vlastnosti jednotlivých materiálových fází, a to se zřetelem na jejich prostorové uspořádání.

Optimální mikrostrukturu kompozitu lze najít například výběrem z množiny strukturně příbuzných vzorků na základě experimentálních měření požadovaných charakteristik [3]. Efektivnějším přístupem se však zdá být Microstructure Sensitive Design (MSD) [3], kdy jsou nejprve vzneseny požadavky na cílové vlastnosti a je určena materiálová báze budoucí suspenze. Následně je (in-silico) generována množina mikrostruktur svázaných s různými technologiemi výroby. Na tyto vzorky jsou aplikovány metody stanovení efektivních vlastností a na jejich základě je vybrána optimální struktura a výrobní postup. Oproti tradičnímu experimentálně založenému postupu je takto možné ušetřit čas a náklady spojené s výrobou a testováním velké sady vzorků.

Pro účely MSD je nezbytná znalost přístupů, které umožní výpočet makroskopického chování kompozitů. Tradiční asymptotické metody pro odhad průměrných vlastností vycházejí z poměrného zastoupení zjednodušené geometrie a topologie jednotlivých fází (např. Mori-Tanaka či Self-consistentní metoda). Tyto metody lze výrazně zpřesnit za předpokladu znalosti skutečné struktury materiálu [4]. Takto zpřesněné metody nejčastěji aplikují numerické modely se specifickými okrajovými podmínkami na určitý vzorek mikrostruktury [5, 6, 4]. Informace o mikrostruktuře jsou většinou získávány pomocí mikroskopů daného materiálu. Vycházet při stanovování efektivních vlastností z celého mikroskopu je výpočetně velmi náročné, z tohoto důvodu



Obrázek 1: Mikrostruktury uvažovaných materiálů: (a) médium s rovnoměrnou distribucí disků, (b) pískovec (bílá barva= póry), (c) hliníková pěna (černá barva= póry)

se většina prací omezuje na komprimovaný, statisticky reprezentativní vzorek mikrostruktury

Předložená bakalářská práce se zabývá možností komprimace mikrostrukturální informace a její následné rekonstrukce s využitím Wangových dlaždic a stochastického dláždění. Původní koncept dláždění, detailně popsany v kapitole 2, byl představen matematikem Hao Wangem jako nástroj pro rozhodování o možnosti algoritmizace matematických problémů [7, 8]. Do dnešní doby však našel uplatnění i v jiných vědních oborech [9]. Výhoda Wangova dláždění spočívá ve schopnosti vytvořit nekonečně velkou neperiodickou strukturu z konečné sady dlaždic [10, 11]. Této vlastnosti se úspěšně využívá například v oblasti počítačové grafiky ke generování přirozeně vyhlížejících textur renderovaných plošných oblastí [9].

V případě mikrostruktury je možné aplikací Wangova dláždění získat dvoudimenzionální reprezentativní vzorky libovolných velikostí (tento postup lze rozšířit i na tři dimenze [12]). Narozdíl od tradičního přístupu k reprezentaci mikrostruktury založeného na periodické buňce PUC [4] není nutné v případě Wangova dláždění zavádět tak striktní předpoklady, jako je periodicitu, a je tedy možné lépe reprezentovat obecnější náhodné struktury. Mikrostrukturální informace je obdobně jako v [4] kvantifikována pomocí statistických deskriptorů, jmenovitě dvoubodové pravděpodobnostní funkce.

Bakalářská práce je členěna následovně: První kapitola je věnována statistickým deskriptorům prostorového uspořádání jednotlivých fází kompozitu. Ve druhé kapitole je představen koncept Wangových dlaždic a stochastického dláždění. Třetí kapitola se věnuje automatické tvorbě dlaždic ze zadaných částí mikrogramů referenčního média. V závěrečné kapitole jsou shrnuty důležité výsledky a je nastíněno možné pokračování této práce.

V práci jsou použity tři materiálové struktury (viz obr. 1). První strukturu tvoří rovnoměrně distribuované kruhové disky. Další dvě mikrostruktury představují vzorky pískovce a hliníkové pěny. Zatímco v prvním případě se jedná o uměle vytvořenou mikrostrukturu, zbylé dvě jsou záznamem reálného materiálu získaného magnetickou rezonancí a skenováním.

1 Statistický popis mikrostruktury

Stanovení efektivních vlastností materiálu na rozsáhlých mikrografech je výpočetně náročné a časově neefektivní [2, 4, 3]. V případě mikrostrukturálních materiálů je tedy výhodné pracovat s reprezentativním vzorkem RVE (Representative Volume Element), který je co nejmenší, ale zároveň natolik velký, aby obsahoval veškerou mikrostrukturální informaci. Tento postup je možný pouze za předpokladu ergodicity zkoumaného média [2, 4].

Uvažujme soubor mikrogramů S daného média a jednu jeho realizaci α s pravděpodobností výskytu $p(\alpha)$. Pak jakákoli průměrná informace $\overline{F(\mathbf{x})}$, $\mathbf{x} \in \mathbb{R}^2$, na dané množině $\mathbb{R}^2 \times S$ je získána průměrem přes celý soubor S jako

$$\overline{F(\mathbf{x})} = \int_S F(\mathbf{x}, \alpha) p(\alpha) d\alpha. \quad (1)$$

Hypotéza ergodicity předpokládá, že objemový průměr funkce $F(\mathbf{x}, \alpha)$ přes celou oblast V

$$\widetilde{F(\mathbf{x}, \alpha)} = \frac{1}{|V|} \int_V F(\mathbf{x} + \mathbf{y}, \alpha) d\mathbf{y} \quad (2)$$

je nezávislý na výběru α a shodný s průměrem funkce $\overline{F(\mathbf{x}, \alpha)}$ přes celý soubor mikrogramů S , tedy

$$\overline{F(\mathbf{x})} = \widetilde{F(\mathbf{x}, \alpha)} \quad (3)$$

pro $V \rightarrow \infty$.

Aby byla podmínka (3) splněna i pro ohraničené oblasti, musí být vzorek α dostatečně veliký a reprezentativní. To je automaticky splněno pro periodické médium [2], kdy místo průměru přes celé $V \rightarrow \infty$ je možno uvažovat průměr přes jednu periodickou část Y . Poté lze výraz (2) přepsat na

$$\widetilde{F(\mathbf{x}, \alpha)} = \frac{1}{|Y|} \int_Y F(\mathbf{x} + \mathbf{y}, \alpha) d\mathbf{y}. \quad (4)$$

Z výše uvedeného lze usuzovat, že i libovolnou funkci $F(\mathbf{x})$ pro nekonečné, neperiodické médium lze aproximovat průměrem přes soubor mikro-

grafů $\overline{F(\mathbf{x}, \alpha)}$, který v případě ergodického média lze stanovit jako průměr $\overline{F(\mathbf{x}, \alpha)}$ přes reprezentativní vzorek.

V našem případě takový vzorek vzniká Wangovým dlážděním pomocí dlaždic uspořádaných v množině, jejíž vlastnosti jsou nastaveny tak, aby bylo možné zrekonstruovat libovolně velký vzorek V .

Aby bylo možné rozhodnout, zda pomocí Wangových dlaždic rekonstruované médium dostatečně odpovídá médiu referenčnímu, je nutno stanovit deskriptory, pomocí kterých bude možno uspořádání mikrostruktury kvantifikovat. Tyto deskriptory lze rozdělit na dvě základní skupiny, na deskriptory prvního řádu a deskriptory řádů vyšších [3]. Ačkoli v rámci této práce je uvažována především dvoubodová pravděpodobnost (two-point probability function), patřící mezi deskriptory druhého řádu, jsou v krátkosti uvedeny i další mikrostrukturální deskriptory.

Mezi deskriptory prvního řádu se řadí jednobodová pravděpodobnostní funkce S_i (one-point probability function) definovaná na základě charakteristické funkce $\chi_i(\mathbf{x}, \alpha)$ (characteristic function) [2] dané výrazem

$$\chi_i(\mathbf{x}, \alpha) = \begin{cases} 1 & \text{pro } \mathbf{x} \in D_i(\alpha), \\ 0 & \text{v ostatních případech,} \end{cases} \quad (5)$$

kde D_i je oblast příslušející i -té fázi v daném vzorku α . Jednobodová pravděpodobnostní funkce S_i je následně definována jako průměr charakteristické funkce $\chi_i(\mathbf{x}, \alpha)$ přes celý soubor S realizací α jako

$$S_i(\mathbf{x}) = \overline{\chi_i(\mathbf{x}, \alpha)} = \int_S \chi_i(\mathbf{x}, \alpha) p(\alpha) d\alpha \quad (6)$$

a udává, s jakou pravděpodobností se bude fáze i nacházet v místě se souřadnicí \mathbf{x} .

Z definice je patrné, že jednobodová pravděpodobnostní funkce $S_i(\mathbf{x})$ je závislá pouze na hodnotě charakteristické funkce v jednom bodě. Naproti tomu deskriptory vyšších řádů vyžadují jako vstupní informace hodnoty χ_i z více bodů. Nejčastěji se používají deskriptory druhého řádu, s vyšším než druhým řádem deskriptorů totiž narůstá náročnost výpočtu a také složitost interpretace výsledků [3]. Vyjma dále zmiňované dvoubodové pravděpodobnostní

funkce, které je věnována část 1.1, patří do této skupiny lineal path function, která zkoumá, zda přímá spojnice mezi dvěma body leží celá v dané fázi, a 2-point cluster function a s ní spojená 2-point blocking function, které udávají, zda se počáteční a koncový bod daného vektoru nalézá v souvislé oblasti jedné inkluze [3].

1.1 Dvoubodová pravděpodobnostní funkce

Dvoubodová pravděpodobnostní funkce je v diskrétním případě specifickým případem dvoubodové korelační funkce (2-point correlation function) definované výrazem

$$S_{rs}(\mathbf{x}_1, \mathbf{x}_2) = \overline{\chi_r(\mathbf{x}_1, \alpha)\chi_s(\mathbf{x}_2, \alpha)}. \quad (7)$$

Fullwood a kol. v článku [3] uvádí dělení dvoubodové korelační funkce na funkce křížově-korelační a auto-korelační v závislosti na tom, zda jsou zkoumané fáze r a s v S_{rs} různé nebo totožné. Z této definice vyplývá, že dvoubodová pravděpodobnostní funkce r -té fáze je dvoubodovou auto-korelační funkcí udávající s jakou pravděpodobností se dva body $\{\mathbf{x}_1, \mathbf{x}_2\}$ vyskytují ve fázi r . Za předpokladu statisticky homogenního média, kdy

$$\overline{F(\mathbf{x}_1, \mathbf{x}_2)} = \overline{F(\mathbf{x}_1 - \mathbf{y}, \mathbf{x}_2 - \mathbf{y})}, \quad (8)$$

platí, že $S_{rs}(\mathbf{x}_1, \mathbf{x}_2) = S_{rs}(\mathbf{x}_1 - \mathbf{x}_2)$ a výraz (7) se tedy redukuje na

$$S_{rs}(\mathbf{y}) = \overline{\chi_r(\mathbf{x}, \alpha)\chi_s(\mathbf{x} + \mathbf{y}, \alpha)}. \quad (9)$$

Za předpokladu ergodicity média lze výraz (7) přepsat s uvažování RVE na tvar

$$S_{rs}(\mathbf{y}) = \frac{1}{|\Omega|} \int_{\Omega} \chi_r(\mathbf{x})\chi_s(\mathbf{x} + \mathbf{y}) d\mathbf{x}, \quad (10)$$

kde Ω je oblast RVE.

Dále lze odvodit hodnoty auto-korelační funkce $S_{rr}(y)$ pro dva limitní případy, pro $\mathbf{y} = 0$ platí $S_{rr}(\mathbf{y}) = \phi_r$, při $\mathbf{y} \rightarrow \infty$ platí $S_{rr}(\mathbf{y}) = \phi_r^2$, kde ϕ_r značí poměrné zastoupení fáze r .

V druhém případě (tj. $y \rightarrow \infty$) se korelace mezi daty v jednotlivých bodech vytrácí a jedná se tedy o sdruženou pravděpodobnost realizace dvou nezávislých náhodných jevů [2].

Budeme-li uvažovat referenční médium jako statisticky izotropní, to znamená, platí-li

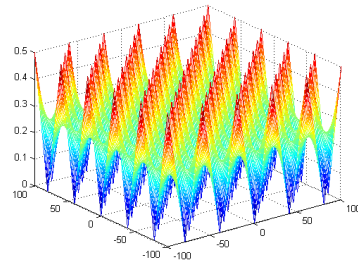
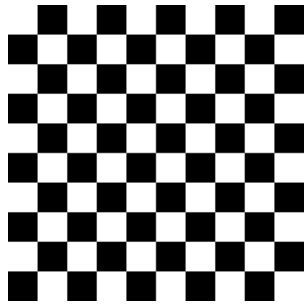
$$\overline{F(|\mathbf{x}_1, \mathbf{x}_2|)} = \overline{F(|\mathbf{x}_1 - \mathbf{x}_2|)} = \overline{F(\mathbf{x}_1 - \mathbf{x}_2)}, \quad (11)$$

je možné dále definovat koherenční vzdálenost r_c (coherence length), kdy $S_{rr}(|\mathbf{y}| \geq r_c) \approx \phi_r^2$ [3].

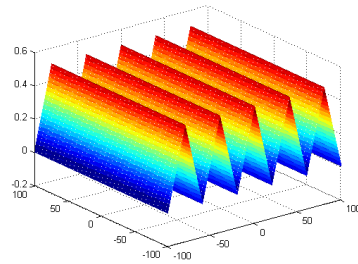
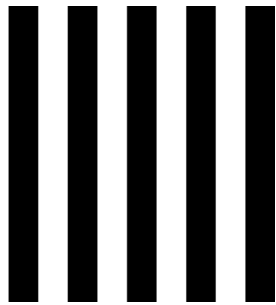
Mikrostruktury uvedené na obr. 2 slouží jako ilustrativní příklad pro interpretaci výsledků dvoubodové pravděpodobnosti vykreslených do grafů. V případě šachovnice, obr. 2(a), je možno vidět silnou periodicitu, se kterou se S_{rr} pohybuje v rozmezí (0.5; 0.0) a která odpovídá střídání jednotlivých políček šachovnice. U dvoufázového laminátu, obr. 2(b), se projevuje výrazná anizotropie, kdy je v podélném směru S_{rr} konstantní a naopak v příčném směru se mění s periodou rovnou dvojnásobku šířky pruhu. Pro náhodně vygenerovaný černobílý šum, obr. 2(c), je pozorovatelný okamžitý pokles z hodnoty ϕ_r pro $S_{rr}(0, 0)$ na hodnotu ϕ_r^2 pro ostatní případy, což znamená, že se v médiu nevyskytuje žádná charakteristická korelační délka a skutečně se jedná o náhodně a izotropně rozmístěné body. Posledním ilustrativním příkladem, obr. 2(d), je médium s disky zmiňované již v úvodu. To se skládá ze vzájemně se nepřekrývajících inkluzí umístěných tak, aby byla vyloučena periodicitu, která by se v dvoubodové pravděpodobnosti zobrazila jako sekundární extrémy grafu (podobně jako u varianty (a) a (b) tohoto obrázku).

1.2 Metody výpočtu dvoubodové pravděpodobnosti

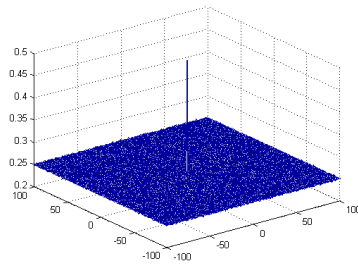
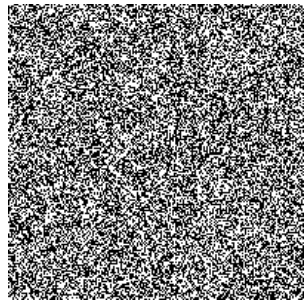
Většina informací o mikrostruktuře materiálu pochází ze záznamů pořízených skenerem (příkladem může být hliníková pěna na obr. 1(c)), elektronovým mikroskopem či magnetickou resonancí. Výsledkem je diskrétní popis materiálu pomocí množiny pixelů s daným rozsahem fází dle příslušných odstínů.



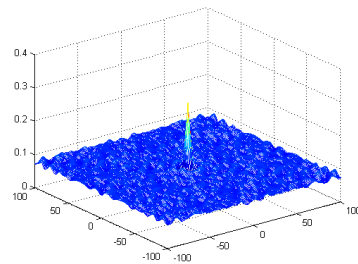
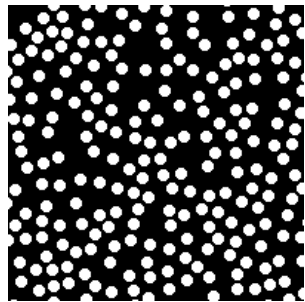
(a)



(b)



(c)



(d)

Obrázek 2: Příklady médií a dvoubodové pravděpodobnosti: (a) šachovnice s kubickou symetrií, (b) ortotropní médium, (c) médium s rovnoměrným rozdělením fází, (d) médium s disky

Jak bylo uvedeno výše, za předpokladu ergodicity média lze funkci $S_{rr}(x)$ stanovit na reprezentativním vzorku RVE, který je uvažován jako periodický. Na základě diskretizace pak lze vyjádřit rovnici (10) pro 2-D záznam jako [2]

$$S_{rr}(kx, ky) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N \chi_r(x, y) \chi_r(x + kx, y + ky), \quad (12)$$

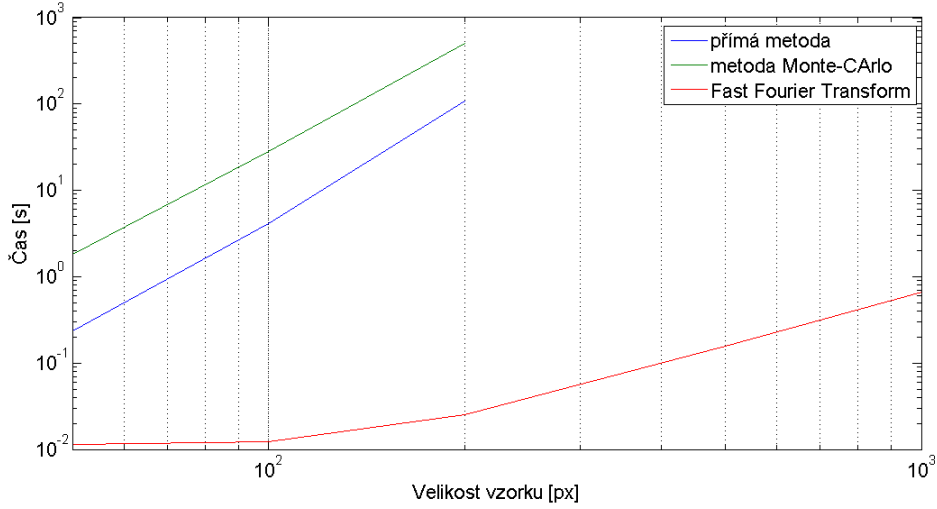
kde M a N je velikost mikrografu v pixelech a diskrétní krok $kx, ky \in \mathbb{Z}$ se pohybuje v rozmezí $\langle -M, M \rangle$, respektive $\langle -N, N \rangle$. Pro výpočet dle (12) je nutné mikrograf periodicky rozšířit o jednu jeho kopii v každém směru, popřípadně lze pro úsporu operační paměti přidat operaci modulo a přepsat (12) na [4]

$$S_{rr}(kx, ky) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N \chi_r(x, y) \chi_r((x + kx)\%M, (y + ky)\%N). \quad (13)$$

Výpočetní algoritmus přímé metody probíhá následovně. Nejdříve se vygenerují všechny vektory (kx, ky) pro $kx \in \langle 0, M \rangle$ a $ky \in \langle -N, N \rangle$. Hodnota kx se pohybuje v intervalu pouze od nuly do M , jelikož je dvoubodová pravděpodobnostní funkce uvažována jako sudá (platí $S_{rr}(x, y) = S_{rr}(-x, -y)$) a její graf je souměrný vzhledem k ose z . Následně algoritmus postupně prochází každý bod mikrografu, v každém bodě vyhodnotí výsledky pro vygenerovaný vektor (kx, ky) a zaznamená do matice případy, kdy se hodnoty pixelů v obou koncových bodech rovnají. Tento algoritmus lze mírně optimalizovat a přidat podmínku kontroly počátečního bodu, kdy jsou vynechány všechny body, které nenáleží fázi r . Z původního počtu kroků $O((MN)^2)$ [2] je tak nutno provádět pouze $O(\phi_r(MN)^2)$ operací.

Přímý postup je i přes zmíněnou optimalizaci výpočetně náročný. Možným řešením je adaptace metody Monte-Carlo, kdy namísto kontroly všech výskytů vektoru (kx, ky) je předem stanoven počet pseudonáhodných realizací $n' < MN$, například poměrem $\rho = \frac{n'}{MN}$. Pravděpodobnost $S_{rr}(kx, ky)$ je pak přibližně spočítána jako poměr realizací splňujících podmínku obou bodů ve fázi r n' ku všem realizacím ρMN ,

$$S_{rr}(kx, yy) = \frac{n'}{\rho MN}. \quad (14)$$



Obrázek 3: Porovnání metod výpočtu dvoubodové pravděpodobnostní funkce na architektuře Intel(R) Core(TM)2 Duo CPU T5870@2,00GHz, 4,00GB RAM, Windows 7 64bit, Matlab R2011b

Pro výpočet metodou Monte-Carlo je nutno provést $O(\rho(MN)^2)$ operací, jejichž počet by šel dále snížit jako u výše uvedené přímé metody na $O(\phi_r \rho(MN)^2)$. Nevýhodou je předem vnesená nepřesnost vlivem aproximace, kdy přesného řešení je dosaženo při $\rho = 1$ a za předpokladu, že žádný vektor nebude umístěn více než jednou do konkrétní pozice. Oproti přímé metodě vychází metoda Monte-Carlo také časově náročnější (viz obr. 3), což je dáno především náročností generátoru náhodných čísel. Pokud bychom však stanovovali S_{rr} pro více mikrogramů, bylo by možné vygenerovat pseudonáhodné realizace pouze jednou, čímž by se časová náročnost snížila.

Uvědomíme-li si speciální tvar integrální rovnice 10 a regulární diskretizaci zkoumaného média, naskytuje se možnost využít k řešení diskretní Fourierovu transformaci (DFT, Discrete Fourier Transform) [2]. Fourierova transformace je integrální transformace, hojně využívaná především v elektotechnice při zpracování signálů, která převádí funkce času $f(t)$ na funkce frekvence $\tilde{f}(\omega)$.

$$\mathcal{F}(f(t)) = \tilde{f}(\omega) \quad (15)$$

Detailněji je Fourierova Transformace představena v příloze C.

Výhodou DFT je výrazné zjednodušení některých matematických operací ve Fourierových obrazech. S ohledem na tvar rovnice (10) se jedná o zjednodušení vztahu pro korelaci dvou funkcí

$$(f \star g)(\mathbf{y}) = \int f^*(\mathbf{x})g(\mathbf{x} + \mathbf{y}) d\mathbf{x}, \quad (16)$$

kde $f^*(x)$ značí funkci komplexně sdruženou s funkcí $f(x)$. Ve Fourierových obrazech lze psát

$$\mathcal{F}\{(f \star g)(\mathbf{y})\} = \mathcal{F}^*\{f(\mathbf{x})\}\mathcal{F}\{g(\mathbf{x})\}. \quad (17)$$

Rovnici (10) je pak možno psát ve tvaru [2]

$$S_{rs}(kx, ky) = \mathcal{F}^{-1}\{\mathcal{F}^*\{\chi_r(x, y)\}\mathcal{F}\{\chi_s(x, y)\}\}. \quad (18)$$

Při využití DFT je třeba provést $O(MN \log(MN))$ operací [2], což je výrazně méně než v obou předchozích způsobech výpočtu.

1.3 Shrnutí

V této kapitole byl zaveden pojem ergodicity média. Ve stručnosti pak byly diskutovány mikrostrukturální statistické deskriptory a detailněji byla představena dvoubodová pravděpodobnostní funkce S_{rs} . Tento deskriptor je dále použit ke kvantifikaci podobnosti rekonstruovaného média a média referenčního. Bylo ukázáno, že přímý výpočet dvoubodové pravděpodobnostní funkce je výpočetně náročný. Jako řešení efektivního stanovení dvoubodové pravděpodobnostní funkce byla implementována metoda založená na diskrétní Fourierově transformaci.

2 Komprimovaná reprezentace mikrostruktury

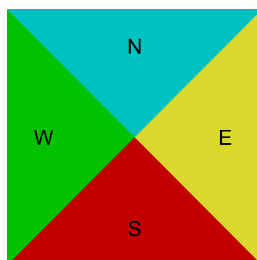
Pro predikci vlastností materiálu ze znalosti jeho mikrostruktury je nutné najít její efektivní reprezentaci. Pro komprimaci a následnou rekonstrukci mikrostruktury lze využít několik postupů. Nejpoužívanější přístup je založen na principu jednotkové periodické buňky (PUC, Periodic Unit Cell) [2, 9]. PUC musí obsahovat veškerou mikrostrukturální informaci, tj. musí být reprezentativním vzorkem, tzv. RVE, popsáním v předchozí kapitole. Médium pak při rekonstrukci vzniká periodickým kopírováním PUC. Nevýhodou tohoto postupu je vznik výrazně periodické struktury, která do značné míry určuje makroskopické chování kompozitu [9].

Tato práce se věnuje dalšímu možnému přístupu založenému na Wangově dláždění. Oproti PUC konceptu tento přístup umožňuje dosáhnout rekonstruované struktury bez viditelnějších artefaktů či periodicity, a to již s relativně malým počtem dlaždic [11].

2.1 Komprimace založená na Wangových dlaždicích

Jak již bylo zmíněno, princip dláždění roviny pomocí Wangových dlaždic umožňuje vytvořit neperiodickou strukturu za pomoci poměrně malé množiny dlaždic. Wangovy dlaždice jsou definovány jako čtverce, jejichž hrany jsou opatřeny kódem nebo barvou [7]. Při tvorbě výsledné struktury jsou pak dlaždice skládány tak, aby byla splněna podmínka kompatibility na všech hranách, což v případě reprezentace mikrostruktury znamená, že na hranách nedochází k nespojitosti jednotlivých fází. Wangovy dlaždice nemohou být při dláždění rotovány ani zrcadleny [7].

Dlaždice jsou pojmenovány po matematiku Hao Wangovi, který v roce 1961 představil tento koncept podobný dominu [7]. Wang poukazuje na podobnost rozhodování, zda lze pro určitý matematický problém najít algoritmus řešení sestávající se z konečného počtu kroků, s problémem, zda s daným souborem dlaždic lze spojitě vydláždít prostor [7, 8]. Jeho práce pak byla dále rozpracována a do současnosti našla uplatnění v rozličných oborech



Obrázek 4: Wangova dlaždice s označením hran

lidské činnosti, od počítačové grafiky, kdy je s výhodou využívána pro tvorbu přirozeně vyhlížejících textur, přes biologii až po modelování jakéhokoli Turingova stroje [11, 9, 8].

Pro potřeby komprimace a rekonstrukce mikrostruktury však zůstává rozhodující schopnost vytvářet spojitou morfologii, která je dostatečně náhodná a neperiodická [10], a to i z malého počtu dlaždic. Z pohledu statistického popisu mikrostruktury není podmínkou, aby každá dlaždice obsahovala celou mikrostrukturální informaci jako je tomu v případě [4]. V [3] je zaveden pojem statistického vzorku SVE (Statistical Volume Element) definovaného tak, že průměr integrálů libovolné funkce přes soubor SVE je roven integrálu přes RVE. Jinak řečeno, není požadováno, aby jednotlivý vzorek SVE obsahoval veškerou mikrostrukturální informaci, postačí, když bude tato informace obsažena v celém souboru, tj. ve všech uvažovaných Wangových dlaždicích. Tento postup však a priori vnáší podmínku ergodicity média [3].

2.2 Stochastické dláždění

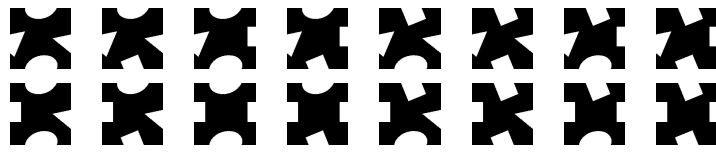
Uvažujme n_v kódů na svislé hraně dlaždice a n_h na hraně vodorovné. Pro větší přehlednost označme jednotlivé hrany dlaždice dle světových stran W, N, E a S (viz obr. 4). Plný soubor Wangových dlaždic, tj. obsahující veškeré kombinace kódů, se pak skládá z $n_v^2 n_h^2$ různých dlaždic. Aby bylo dláždění proveditelné, stačí, aby soubor dlaždic obsahoval jednu dlaždici pro každou N-W kombinaci. V tomto případě ale ztrácí algoritmus svoji stochastickou povahu. Aby

bylo dláždění náhodné, je třeba alespoň dvou různých dlaždic pro každou N-W kombinaci. Použitelný set tedy musí obsahovat nejméně $2n_v n_h$ různých dlaždic.

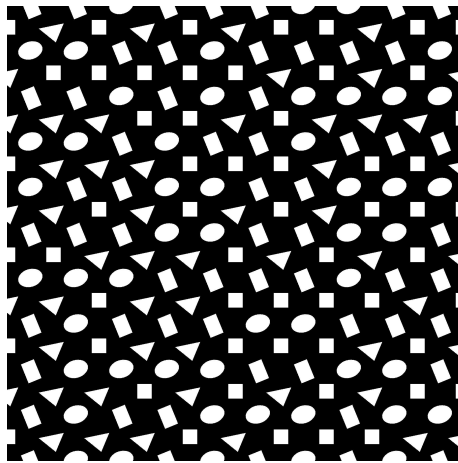
Postup dláždění je založen na Cohen-Shade-Hiller-Deussen (CSHD) algoritmu představeném v [11]. Barevné kódy z původního konceptu jsou nahrazeny celočíselnými kódy. Algoritmy zmiňované v postupu dláždění uvedeného níže jsou uvedeny v příloze A.

V prvním kroku je seznam dlaždic *TileList* seřazen vzestupně dle W a N kódu. Poté je tento seznam analyzován a do matice *CodeTable* je zapísán počet dlaždic každé kombinace N-W kódů (dané pozicí v matici) do *CodeTable(W,N).number* a index první takové dlaždice v *TileList* do *CodeTable(W,N).begin*. Na základě zadaných rozměrů rekonstruované části roviny *nVer* a *nHor* je vytvořena prázdná matice *Plane*. Postupně z levého horního rohu jsou do této matice náhodně vybírány jednotlivé dlaždice. V každém kroku se nejprve přečte kód na východní-E hraně dlaždice o pozici vlevo (slouží jako západní-W kód pro hledanou dlaždici) a jižní-S kód na dlaždici o pozici výš (tvoří severní-N kód pro novou dlaždici). V případě absence těchto informací (tj. v prvním sloupci a prvním řádku) je chybějící kód náhodně vygenerován. Na základě stanovené N-W kombinace je pak vybrána dlaždice z intervalu $\langle CodeTable(W,N).begin; CodeTable(W,N).begin + CodeTable(W,N).number - 1 \rangle$. Výběr je prováděn s rovnoměrným rozdělením pravděpodobnosti. Jak již bylo zmíněno, tento postup je smysluplný za předpokladu, že pro každou N-W kombinaci existují minimálně dvě dlaždice. V průběhu dláždění se pro úsporu paměti operuje pouze s adresou jednotlivých dlaždic v *TilesList* a na závěr jsou tyto adresy nahrazeny grafickou reprezentací a výsledek uložen.

Na obr. 5 je uveden příklad setu 16 dlaždic a vydlážděné části roviny. Je patrné, že díky podmínce kompatibility nedochází na hranách k nespojitostem rekonstruované textury. Náhodné rozmístění dlaždic je pak znázorněno na obr. 6, kde každé dlaždici odpovídá jedna barva. Dále je z obr. 6 patrné, že z důvodu relativně malého počtu dlaždic v setu dochází k jejich lokálnímu

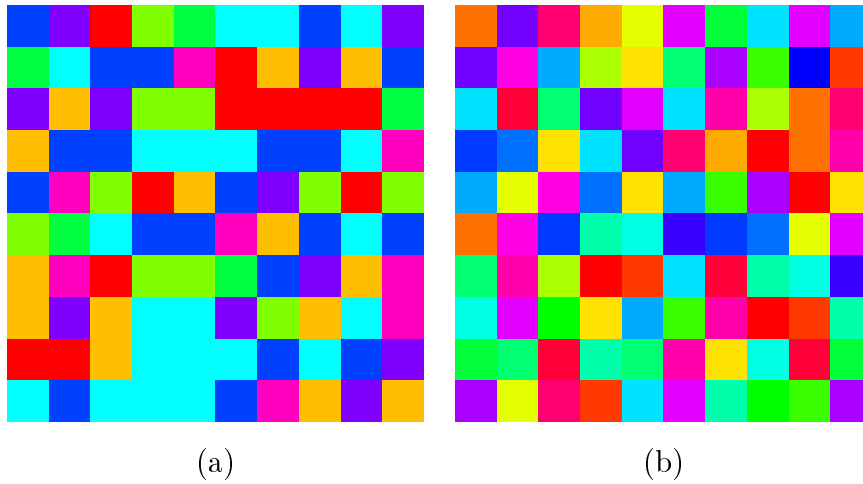


(a)



(b)

Obrázek 5: Příklad (a) Wangových dlaždic (b) a rekonstruované části roviny pro $nVer=10$, $nHor=10$



Obrázek 6: Rozmístění dlaždic v matici 10x10 pro (a) 8 dlaždic v setu a (b) 27 dlaždic v setu

opakování.

Pokud bychom se těmto lokálním periodicitám chtěli vyhnout, bylo by nutné upravit algoritmus dláždění. Například tak, aby vyjma N-W kombinace algoritmus zahrnoval více informací o okolí pokládané dlaždice. Při výběru dlaždice je pak možné dodat podmínku, která kontroluje, zda náhodně vybraná dlaždice není shodná s dlaždicí v jejím nejbližším okolí. Tato úprava ovšem limituje náhodnost při výběru dlaždic a je tedy otázka, zda lokální zlepšení nepovede ke vzniku globálních artefaktů. Dále je nutno zajistit, aby velikost kontrolovaného okolí nepřesáhla počet dlaždic pro N-W kombinaci, to znamená, že v případě pouze tří dlaždic pro každou N-W kombinaci má smysl kontrolovat pouze dvě dlaždice, např. sousední vlevo a nad. Druhou možností je do podmínky přidat kontrolu opakování, která v případě, že dojde k zacyklení (tj. omezení překročí počet dlaždic pro danou kombinaci), pustí algoritmus dál po dosažení určitého počtu opakování výběru. Úprava výběru dlaždice je uvedena v alg. 10. Na obr. 7 je pak znázorněn vliv úpravy CSHD algoritmu. V případě neupravené verze (obr. 7(a,c)) je patrné shlukování identických dlaždic. Naopak u upraveného CSHD algoritmu (obr. 7(b,d)) je tento jev omezen. Jistou daní za tuto úpravu je již zmíněné omezení náhod-

nosti. V případě obr. 7(b,d) se toto omezení projevilo vznikem opakujících se dvojic.

Algoritmus 1: Upravený výběr dlaždice pro omezení vzniku lokálních periodicit

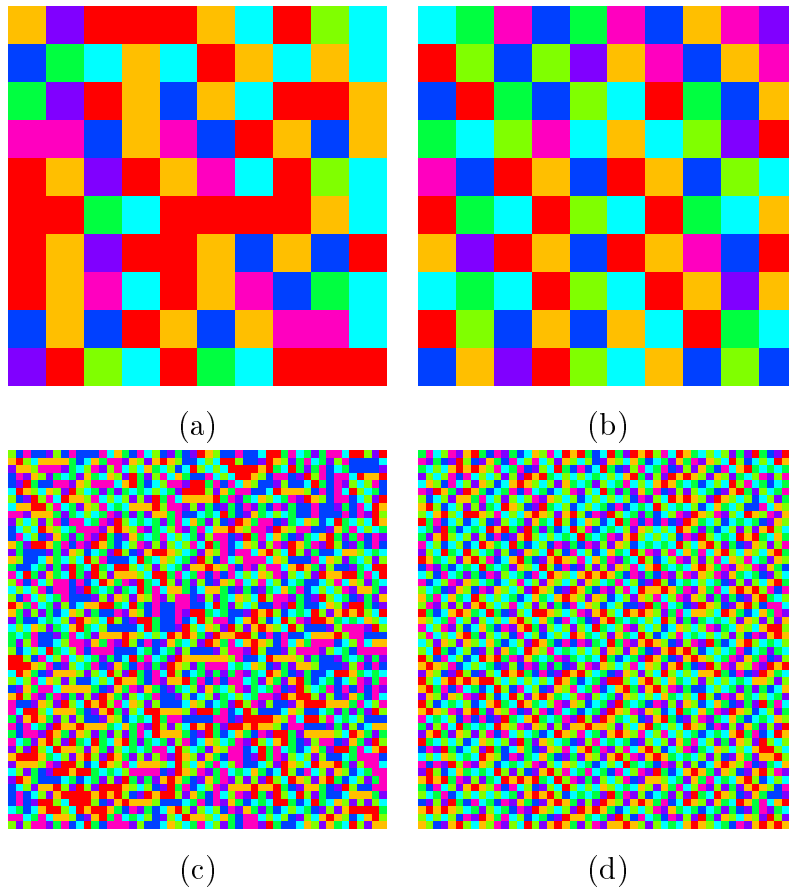
```

1 rep=0;
2 while (rep<=repmax)
3     a=CodeTable(W,N).begin + floor( CodeTable(W
        ,N).number * rand(1));
4     if ((i==1)&&(j==1))
5         rep=repmax+1;
6     elseif ((i==1)&&(a~=Plane(i,j-1)))
7         rep=repmax+1;
8     elseif ((j==1)&&(a~=Plane(i-1,j)))
9         rep=repmax+1;
10    elseif ((i~=1)&&(j~=1)&&(a~=Plane(i,j-1))
        &&(a~=Plane(i-1,j)))
11        rep=repmax+1;
12    end;
13    rep=rep+1;
14 end;
15 Plane(i,j)=a;

```

2.3 Shrnutí

V této kapitole byl představen koncept Wangova dláždění a některé jeho aspekty ve smyslu aplikace na kompresi a rekonstrukci mikrostruktury. Dále byl detailně popsán CSHD algoritmus dláždění z [11]. Byla navržena možná úprava omezující vznik lokálních periodicit. Wangovým dlaždicím a jejich tvorbě se věnuje kapitola 3. Vliv počtu dlaždic na navrženou úpravu CSHD algoritmu je rozebrán v poslední kapitole.



Obrázek 7: Wangovo dláždění roviny 10x10 a 100x100 ze setu obsahujícího 8 dlaždic pomocí (a,c) CSHD algoritmu, (b,d) upraveného CSHD algoritmu

3 Konstrukce Wangových dlaždic

V přechodí kapitole byl představen koncept Wangova dláždění pro rekonstrukci mikrostruktury, jednotlivým dlaždicím a jejich konstrukci je věnována tato kapitola.

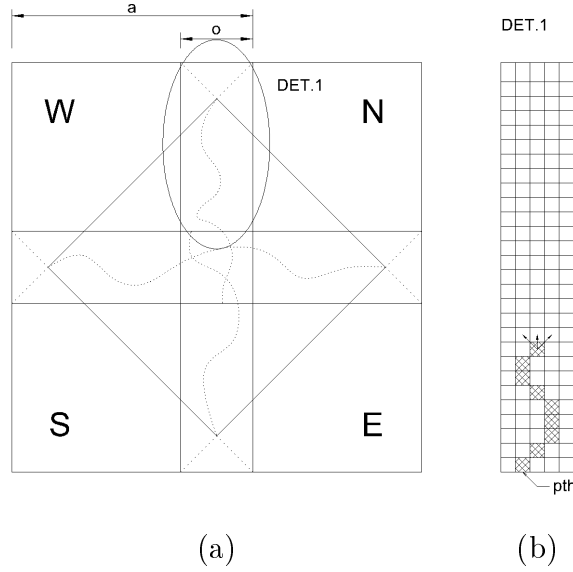
Základním předpokladem pro fungování Wangova dláždění je zajištění návaznosti dlaždic na shodně označených hranách. Tato podmínka tedy významně ovlivňuje návrh jednotlivých dlaždic. Cohen a kol. v práci [11] rozlišují dva základní přístupy k návrhu dlaždic. První přístup je interaktivní tvorba dlaždic, kdy autor musí explicitně zajistit kompatibilitu na hranách opatřených stejným kódem. Do této skupiny lze zařadit i optimalizační metody, které konstruují dlaždice na základě minimalizace odchylky ve statistických deskriptorech mezi dlaždicí a referenčním médiem. Příkladem tohoto přístupu může být [9].

Druhým přístupem dle [11] je automatický návrh dlaždic založený na Image Quilting algoritmu představeném v [1]. Ten je také dále uvažován v rámci této bakalářské práce.

3.1 Konstrukce dlaždic pomocí algoritmu Image Quilting

Při automatickém návrhu dlaždic dle postupu uvedeném v [11] vzniká každá dlaždice oříznutím čtyř vzájemně se překrývajících vzorků, viz obr. 8. Kompatibilita na hranách je zajištěna diagonálním řezem vzorků W,N,E a S, které zároveň slouží jako kód hrany výsledné dlaždice. Tímto přístupem je možno zajistit spojitost na hranách dlaždic i pro složité mikrostruktury, které by v případě interaktivního přístupu k návrhu dlaždic (např. pomocí simulovaného žihání) byly výpočetně velmi náročné.

Relativní jednoduchost zajištění kompatibility na hranách je ale vykoupěna nutností spojit vzorky W až S bez viditelných artefaktů. Elegantní metodu, jak bezešvě sdružit dva vzorky, představili Efros a Freeman v [1]. Jejich postup spočívá v nalezení spojitě cesty v překryvu dvou vzorků, která



Obrázek 8: Schéma automatické konstrukce Wangových dlaždic (a), detail výběru vektoru pth (b)

minimalizuje součet odchylek pixelů. Algoritmus tohoto řešení je následující.

Nejprve je vytvořena submatice přesahu A_o pro vzorek A, respektive B_o pro vzorek B, z těchto submatic je stanovena matice lokálních odchylek e dle vztahu

$$e(i, j) = (A_o(i, j) - B_o(i, j))^2, \quad (19)$$

kde i, j jsou indexy prvků matice. Jeden prvek matice přísluší jednomu pixelu. Dle rovnice (20) je poté vypočtena matice kumultaivní chyby E

$$E(i, j) = e(i, j) + \min(E(i - 1, j - 1), E(i - 1, j), E(i - 1, j + 1)). \quad (20)$$

Jelikož je matice E sestavena postupně z levého horního rohu, další krok algoritmu musí nutně postupovat zdola. Nejprve je nalezeno minimum na posledním řádku matice E a jeho pozice je zaznamenána do vektoru pth . Na dalších řádcích (tj. vždy o jeden výš) je pak minimum vybíráno pouze v rozsahu jednoho pixelu vlevo a vpravo od minima v předchozím řádku, viz obr. 8(b). Takto je možno sestavit vektor pth obsahující polohu pixelu



(a)

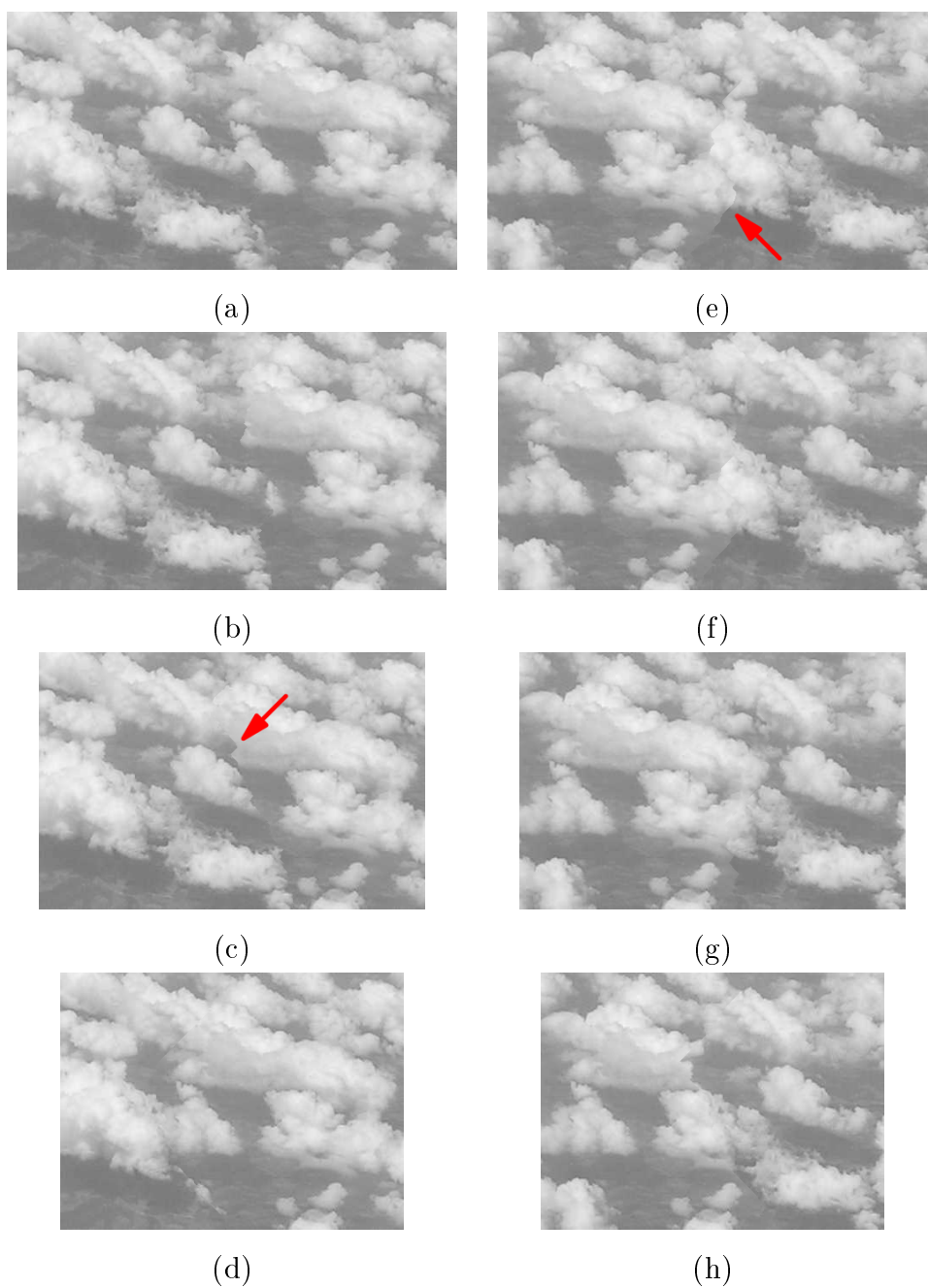
(b)

Obrázek 9: Vstupní vzorky pro obr. 10

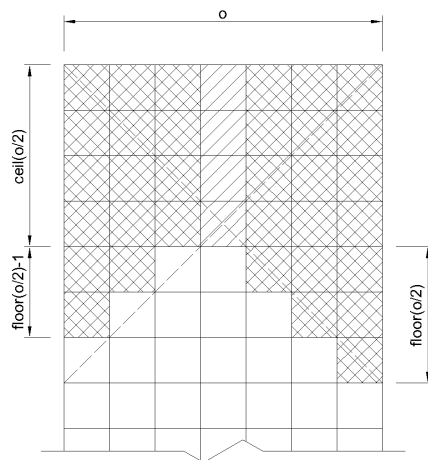
v řádku, který se nachází na spojitě cestě s příspěvkem nejmenší chyby. Posledním krokem je syntéza obou vzorků na základě vektoru pth , kdy pro všechny body na řádku i od $j=0$ do pozice $j=pth(i)$ včetně jsou vybrány pixely ze vzorku A, od pozice $j=pth(i)$ až do $j=size(e,2)$ pak ze vzorku B.

Na obr. 10 jsou znázorněny výsledky Image Quilting algoritmu pro různé hodnoty přesahu a pořadí vstupních vzorků z obr. 9. Zároveň je u každého uveden součet kumulované chyby E počítané podél pth . Dále jsou na obr. 10 vyznačeny artefakty u výsledků s největší kumulovanou chybou. Je patrné, že výsledek záleží nejen na velikosti přesahu vzorků, odkud je pth vybírána, ale i na míře podobnosti vstupních vzorků. Pro příklad uvedený na obr. 10 vycházela při záměně pořadí vzorků menší kumulativní chyba, přičemž změna pořadí navíc ovlivnila i optimální přesah. Hodnotu přesahu lze optimalizovat, přesto u jednoduchých mikrostruktur uvažovaných v této práci lze stanovit přesah expertním odhadem tak, aby byl přibližně větší nebo roven charakteristické délce mikrostrukturálních komponent (v uvedeném ilustrativním případě by se jednalo o šířku a výšku mraků). Zároveň je nutné si uvědomit, že spojnice vzorků pth může probíhat pouze přímo nebo pod úhlem 45° a nemůže se vracet. Toto omezení je daní za poměrně snadný způsob nalezení optimálního vektoru pth .

Aby bylo možné využít tento algoritmus pro tvorbu dlaždic, jak je zobrazeno na obr. 8, je nutné jej dále rozšířit. Především je třeba zajistit, aby



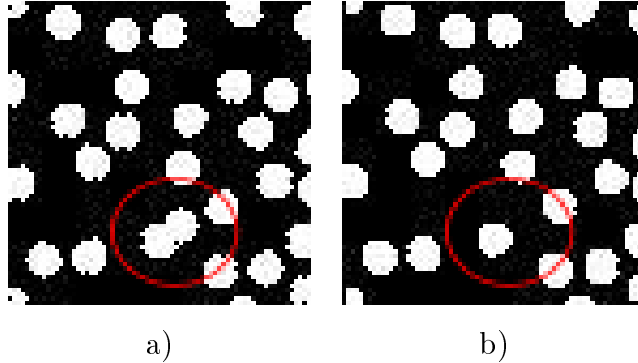
Obrázek 10: Výsledky bezešvého spojení dvou vzorků z obr. 9 pomocí Image Quilting algoritmu. Vždy je uvedeno pořadí obrázků z obr. 9 při syntéze, přesah a kumulovaná chyba. (a) 1-2, 75px, 58172; (b) 1-2, 100px, 63479; (c) 1-2, 150px, 74469; (d) 1-2, 200px, 43246; (e) 2-1, 75px, 102294; (f) 2-1, 100px, 34258; (g) 2-1, 150px, 16830; (h) 2-1, 200px, 34460



Obrázek 11: Schéma penalizace prvků matice e . Do obousměrně šrafovaných pixelů vnesena chyba 10^{20} , do jednosměrně šrafovaných pixelů pak 10^{10}

spoj pth procházel vrcholy budoucí dlaždice. Toho lze dosáhnout penalizací prvků matice e , viz obr. 11, kdy je do určité oblasti předem vnesena vysoká chyba, která zde zamezí výběru cesty pth . Tato penalizace je nesymetrická v důsledku postupu při spojování vzorků A a B, kdy pixely nacházející se na pth jsou vybírány z A. Jinými slovy, aby se alespoň jeden pixel z A, resp. B, nacházel na diagonále (na obr. 8 znázorněno přerušovanou čarou), musí být penalizace asymetrická.

Výše uvedený postup byl implementován v prostředí Matlab následujícím způsobem. Nejprve jsou sloučeny vzorky W a N, na jejich přesah je aplikována penalizace ve formě procedury *ErrorMask* a je stanovena kumulativní chyba pomocí *CumulativeError*. Následně je *MinErrorPath* procedurou stanoven vektor pth . Na základě tohoto vektoru jsou vzorky W a N spojeny procedurou *ImageQuilting*. Stejný postup je použit i v případě spojení E a S vzorků, jediný rozdíl je v aplikaci penalizace *ErrorMask* na spodní část přesahu obou vzorků. V případě spojování WN a ES je nutno tyto mezivýsledky otočit o 90° proti směru hodinových ručiček, jelikož je procedura *ImageQuilting* napsána pro spojení dvou vzorků uspořádaných horizontálně vedle sebe, a aplikovat *ErrorMask* na horní i dolní část přesahu. Výsledek WNES je následně zpět



Obrázek 12: Srovnání Image Quiltingu (a) a jeho upravené verze (b)

otočen o 90° po směru hodinových ručiček.

Z výsledného vzorku WNES je nutno vyříznout Wangovu dlaždici. Prostředí Matlab umožňuje pouze ortogonální výřez, a jelikož se dále nepodařilo explicitně určit, kde se budou nacházet rohy budoucí dlaždice po otočení vzorku WNES příkazem *imrot*, byl napsán vlastní algoritmus pro otočení a ořez dlaždice. Detailněji je tento algoritmus popsán v příloze B.

Vyjma omezení samotného spoje na lomenou čáru, která je buď přímá, nebo pod úhlem 45° , má uvedený Image Quilting algoritmus nevýhodu v případě přesně definovaných tvarů inkluzí, kdy umožňuje vznik nepůvodních tvarů spojením dvou inkluzí na hraně *pth* (viz obr. 12(a), kde došlo k penetraci dvou disků). Navrženým řešením pro dvoufázové médium je úprava výrazu stanovení lokální chyby (19) na

$$e(i, j) = \begin{cases} 1 & \text{pokud } A_o(i, j) = B_o(i, j) = m, \\ 2 & \text{pokud } A_o(i, j) = B_o(i, j) = i, \\ 3 & \text{v ostatních případech,} \end{cases} \quad (21)$$

kde m je číselná hodnota reprezentující matrici a i číselná hodnota inkluze. Touto úpravou je preferován spoj, který probíhá matricí. Aby tento postup mohl fungovat, musí se jednat o inkluze, které jsou plně oddělené matricí, např. sktruktura obsahující bílé disky či pískovec. Příklad důsledku této úpravy je zřejmý z obr. 12. Ačkoli je úprava uvedena pro dvoufázový materiál, obdobně je možné postupovat i v případě vícefázového prostředí.

3.2 Shrnutí

V této kapitole byl představen algoritmus Image Quilting a jeho implementace pro automatickou konstrukci Wangových dlaždic. Pro eliminaci nevýhod tohoto algoritmu pro dvoufázové materiály s přesně definovanými tvary inkluzí, kdy mohou vznikat nepůvodní tvary, je navržena jeho modifikace.

Závěr

V předložené bakalářské práci byla řešena komprimace materiálové mikrostruktury za pomoci Wangova dláždění. Pro kvantifikaci prostorového uspořádání mikrostruktury byly použity statistické deskriptory a detailně byla představena dvoubodová pravděpodobnostní funkce S_{rs} včetně možnosti jejího výpočtu. Dále byl představen automatický návrh Wangových dlaždic založený na algoritmu Image Quilting.

V rámci práce byly navrženy dvě úpravy dříve představených algoritmů. První úpravou je modifikace Image Quilting algoritmu pro eliminaci vzniku nepůvodních tvarů inkluzí u dvofázových materiálů. Druhá navržená úprava omezuje vznik lokálních periodicit přidáním kontroly okolí při Wangově dláždění.

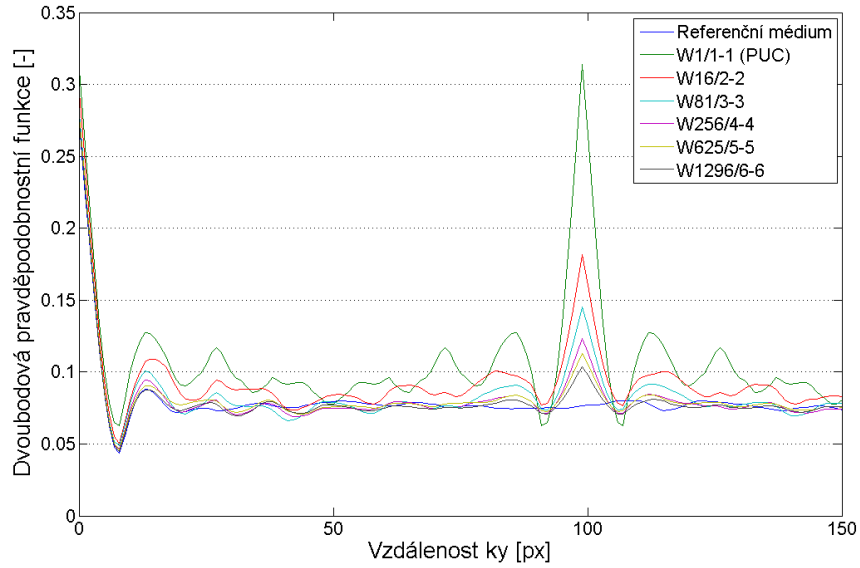
Pro porovnání automatické konstrukce Wangových dlaždic s konstrukcí dlaždic založenou na optimalizačních metodách [9] byla vybrána mikrostruktura s disky (viz obr. 1(a)). Postupně byl vygenerován plný soubor Wangových dlaždic pro počet kódů na hranách $i \in \{1, 2, 3, 4, 5, 6\}$ a byl stanoven průběh $S_{rr}(0, ky)$ na průměru z třiceti realizací dláždění o velikosti 15×15 dlaždic.

Je patrné, že narozdíl od setů dlaždic zkonstruovaných pomocí optimalizačních metod [9] je velikost sekundárního extrému závislá na počtu kódů na hranách a ne na počtu dlaždic v setu. Toto je dáno automatickou konstrukcí dlaždic, kdy i pro $n = i^4$ dlaždic v setu se stále jedná o i opakujících se vzorků.

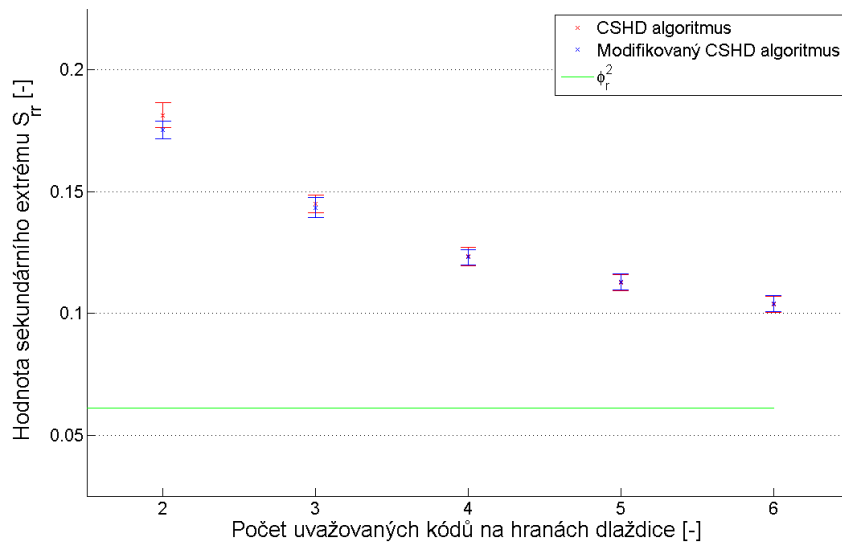
Z obr. 14 je vidět, že modifikovaný CSHD algoritmus snižuje hodnotu sekundárních extrémů především v případě setů obsahujících menší počet dlaždic. U setů s větším počtem dlaždic nedochází k lokálním periodicitám a priori, proto se také vliv úpravy algoritmu vytrácí.

Na obr. 15 jsou uvedeny výsledky rekonstrukce pro další dvě zmiňované mikrostruktury z obr. 1.

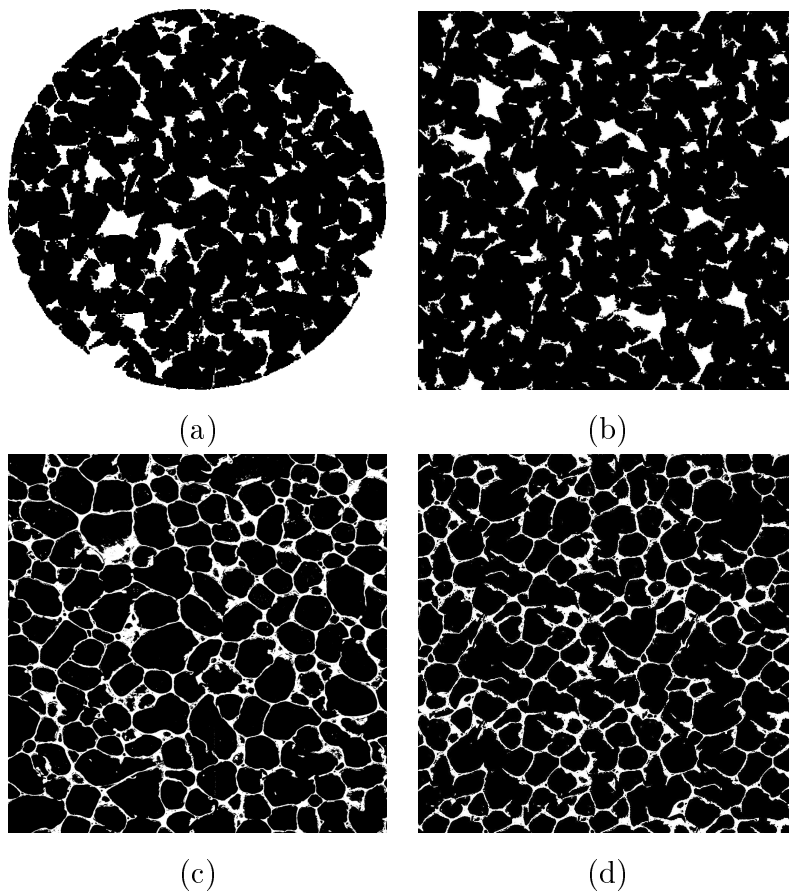
Koncept Wangova dláždění se zdá být vhodnou metodou k efektivní reprezentaci mikrostruktury. Automatická konstrukce založená na Image Quil-



Obrázek 13: Porovnání průběhů funkce $S_{rr}(0, ky)$ pro plné soubory Wangových dlaždic o i kódech na hranách $i \in \{1, 2, 3, 4, 5, 6\}$



Obrázek 14: Porovnání sekundárních extrémů funkce $S_{rr}(0, ky)$ pro plné soubory Wangových dlaždic o i kódech na hranách $i \in \{1, 2, 3, 4, 5, 6\}$



Obrázek 15: Ukázky referenčního (a)(c) a rekonstruovaného média (b)(d):
(a)(b) pískovec, (c)(d) hliníková pěna

tingu však vyžaduje pro redukci sekundárních extrémů funkce S_{rr} velký počet kódů na hranách. Takto velký soubor Wangových dlaždic je však nepřijatelný pro další výpočty termomechanických vlastností.

Možným řešením se zdá být kombinace přístupu využívajícího optimalizačních metod [9] a automatické konstrukce Wangových dlaždic, případně využít Image Quilting ve dvou úrovních, v první, kroku k tvorbě hran, v druhém kroku pak k dodání informace o morfologii vnitřní části dlaždice. Tyto varianty mohou být zkoumány jako pokračování této práce.

Literatura

- [1] A. Efros, W. Freeman, Image quilting for texture synthesis and transfer, Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001) 341–346.
- [2] M. Šejnoha, J. Zeman, Micromechanical analysis of random composites, Vol. 6 of CTU Reports, Czech Technical University in Prague, 2002.
- [3] D. Fullwood, S. Niezgoda, B. Adams, S. Kalidindi, Microstructure sensitive design for performance optimization, Progress in Materials Science 55 (6) (2010) 477–562.
- [4] J. Zeman, Analysis of composite materials with random microstructure, Ph.D. thesis, Czech Technical University in Prague (2003).
- [5] T. Kanit, S. Forest, I. Galliet, V. Mounoury, D. Jeulin, Determination of the size of the representative volume element for random composites: statistical and numerical approach, International Journal of Solids and Structures 40 (13) (2003) 3647–3679.
- [6] V. Šmilauer, Z. Bittnar, Microstructure-based micromechanical prediction of elastic properties in hydrating cement paste, Cement and Concrete Research 36 (2006) 1708–1718.
- [7] H. Wang, Proving theorems by pattern recognition-II, Bell Systems Technical Journal 40 (1) (1961) 1–42.
- [8] H. Wang, Games, logic and computers, Scientific American 213 (5) (1965) 98–106.
- [9] J. Novák, A. Kučerová, J. Zeman, C. Quince, Compressing random microstructures via stochastic Wang tilings, [online; citováno 18. dubna 2012] (2012).
URL <http://arxiv.org/abs/1206.0933>

- [10] R. Berger, Undecidability of the domino problem, Vol. 66, Memoirs of the American Mathematical Society, 1966.
- [11] M. Cohen, J. Shade, S. Hiller, O. Deussen, Wang tiles for image and texture generation, ACM Transactions on Graphics 22 (3) (2003) 287–294.
- [12] K. Culik II, J. Kari, An aperiodic set of Wang cubes, Journal of Universal Computer Science 1 (10) (1995) 675–686.
- [13] B. Hudec, Základy počítačové grafiky, ČVUT, 1993.
- [14] C. Burrus, T. Parks, DFT/FFT and convolution algorithms: Theory and implementation, John Wiley & Sons, Inc., 1991.
- [15] E. W. Weisstein, "Fourier transform" from MathWorld—A Wolfram web resource, [online; citováno 28. dubna 2012] (2012).
URL <http://mathworld.wolfram.com/FourierTransform.html>
- [16] J. Cooley, J. Tukey, An algorithm for the machine calculation of complex Fourier series, Mathematics of Computation 19 (90) (1965) 297–301.

Seznam obrázků

1	Mikrostruktury uvažovaných materiálů	2
2	Příklady médií a dvoubodové pravděpodobnosti	8
3	Porovnání metod výpočtu dvoubodové pravděpodobnostní funkce	10
4	Wangova dlaždice s označením hran	13
5	Příklad Wangových dlaždic a části rekonstruované roviny	15
6	Příklad rozmístění dlaždic	16
7	Porovnání CSHD a modifikovaného CSHD algoritmu	18
8	Schéma automatické konstrukce Wangových dlaždic	20
9	Vstupní vzorky pro obr. 10	21
10	Výsledky bezešvého spojení dvou vzorků	22
11	Schéma penalizace prvků matice e	23
12	Srovnání Image Quiltingu a jeho upravené verze	24
13	Porovnání průběhů funkce $S_{rr}(0, ky)$	27
14	Porovnání sekundárních extrémů funkce $S_{rr}(0, ky)$	27
15	Ukázky referenčního a rekonstruovaného média	28
16	Schéma otočení a ořezu	49
17	Schéma bilineární transformace	50

Seznam algoritmů

1	Upravený výběr dlaždice pro omezení vzniku lokálních periodicit	17
2	Procedura <i>TilingMain</i>	34
3	Procedura <i>TileCreation</i>	38
4	Procedura <i>ECreate</i>	39
5	Procedura <i>CumulativeError</i>	40
6	Procedura <i>ErrorMask</i>	40
7	Procedura <i>MinErrorPath</i>	41
8	Procedura <i>ImageQuilting</i>	42
9	Procedura <i>ImageRotationCrop</i>	43
10	Procedura <i>TwoPointProbability</i>	44

A Implementace v prostředí Matlab

Algoritmus 2: Procedura *TilingMain*

```
1 % Wang Tiles Algorithm
2 tic
3 % inputs
4 %% list of tiles containing codes for each edge in W,N,
   S,E order
5 nVer=6;
6 nHor=6;
7 Overlap=30;
8 TilesList=zeros((nHor*nVer)^2,4);
9 m=1;
10 for i=1:nVer
11     for j=1:nHor
12         for k=1:nVer
13             for l=1:nHor
14                 TilesList(m,:)= [i ,j ,k ,l ];
15                 m=m+1;
16             end;
17         end;
18     end;
19 end;
20 %% Tiles created using TileCreation procedure from
   input images
21 for i=1:size(TilesList,1)
22     imgW=im2bw(imread([num2str(TilesList(i,1)), '.png']))
        ); % possible to add 'v.png'
23     imgN=im2bw(imread([num2str(TilesList(i,2)), '.png']))
        ); % possible to add 'h.png'
24     imgE=im2bw(imread([num2str(TilesList(i,3)), '.png']))
```

```

        ); % possible to add 'v.png'
25  imgS=im2bw(imread([num2str(TilesList(i,4)), '.png']))
        ); % possible to add 'h.png'
26  imgWNES=(TileCreation(imgW,imgN,imgE,imgS,Overlap))
        ;
27  imwrite(imgWNES,[num2str(TilesList(i,1)),num2str(
        TilesList(i,2)),num2str(TilesList(i,3)),num2str(
        TilesList(i,4)), '.jpg'])
28  end;
29  %
30  %% TilesList analysis
31  TilesList=sortrows(TilesList); %
        sorts the tiles by the edgecode
32  nVer=max(max(TilesList(:,1)),max(TilesList(:,3))); %
        nVer- number of vertical possibilities
33  nHor=max(max(TilesList(:,2)),max(TilesList(:,4))); %
        nHor- number of horizontal possibilities
34  % CodeTable(W,N) matrix contains first index in
        TilesList and number of each W-N combination
35  for i=1:nVer
36      for j=1:nHor
37          CodeTable(i,j).begin=0;
38          CodeTable(i,j).number=0;
39      end;
40  end;
41  for i=1:size(TilesList,1)
42      if CodeTable(TilesList(i,1),TilesList(i,2)).begin
        ==0
43          CodeTable(TilesList(i,1),TilesList(i,2)).begin=
            i;
44      end;

```

```

45     CodeTable( TilesList ( i ,1) , TilesList ( i ,2) ) . number=1+
        CodeTable( TilesList ( i ,1) , TilesList ( i ,2) ) . number;
46 end;
47 %
48 %% main tiling
49 repmax=4;    % number of maximal repetition for modified
        CSHD algorithm
50 for l=1:3
51 nR=l*5;      % number of rows in output image
52 nC=l*5;      % number of columns in output image
53
54 for k=1:30   % tiling repeated 30times
55 N=0;        % kódové číslo N hrany
56 W=0;        % kódové číslo W hrany
57 Plane=zeros(nR,nC);
58
59 for i=1:nR
60     for j=1:nC
61         if i==1
62             N=ceil(nHor*rand(1));
63         else
64             N=TilesList(Plane(i-1,j),4);
65         end;
66         if j==1
67             W=ceil(nVer*rand(1));
68         else
69             W=TilesList(Plane(i,j-1),3);
70         end;
71         % original CSHD algorithm
72 %         Plane(i,j)=CodeTable(W,N).begin + floor(
        CodeTable(W,N).number * rand(1));

```

```

73      % modified CSHD algorithm
74      rep=0;
75      while (rep<=repmax)
76          a=CodeTable(W,N).begin + floor( CodeTable(W
              ,N).number * rand(1));
77          if ((i==1)&&(j==1))
78              rep=repmax+1;
79          elseif ((i==1)&&(a~=Plane(i,j-1)))
80              rep=repmax+1;
81          elseif ((j==1)&&(a~=Plane(i-1,j)))
82              rep=repmax+1;
83          elseif ((i~=1)&&(j~=1)&&(a~=Plane(i,j-1))
              &&(a~=Plane(i-1,j)))
84              rep=repmax+1;
85          end;
86          rep=rep+1;
87      end;
88      Plane(i,j)=a;
89      % end of modified CSHD algorithm
90  end;
91 end;
92 %
93 % plane matrix post-processing, codes replaced with
      images from the tile creation algorithm
94 output=[];
95 for i=1:nR
96     A=[];
97     for j=1:nC
98         A=[A, imread( [num2str( TilesList( Plane(i,j),
              1) ) num2str( TilesList( Plane(i,j), 2) )
              num2str( TilesList( Plane(i,j), 3) )

```

```

        num2str( TilesList( Plane(i,j), 4) ) '.jpg' ]
        ) ];
99     end;
100    output=[output;A];
101 end;
102
103 imwrite(output, ['result_', num2str(1), '_', num2str(k, '%
        03i'), '.jpg' ]);
104 end;
105 end;
106 toc

```

Algoritmus 3: Procedura *TileCreation*

```

1 function [ out_img ] = TileCreation( in_imgW, in_imgN,
    in_imgE, in_imgS, in_overlap )
2 %TILECREATION creates final Wang tile using quilting
    algorhitm
3 %
4 overW = in_imgW(:, (size(in_imgW,2)-in_overlap+1):size(
    in_imgW,2) );
5 overN = in_imgN(:, 1:in_overlap);
6 %e=(double(overW)-double(overN)).^2;
7 e=ECreate(overW, overN);
8 e=ErrorMask(e);
9 E=CumulativeError(e);
10 [path, error]=MinErrorPath(E);
11 imgWN=ImageQuilting(in_imgW, in_imgN, in_overlap, path);
12 %
13 overS = in_imgS(:, (size(in_imgS,2)-in_overlap+1):size(
    in_imgS,2) );
14 overE = in_imgE(:, 1:in_overlap);
15 %e=(double(overS)-double(overE)).^2;

```

```

16 e=ECreate(overS , overE );
17 e=flipud(e); % necessary
18 e=ErrorMask(e);
19 e=flipud(e); % necessary
20 E=CumulativeError(e);
21 [path , error]=MinErrorPath(E);
22 imgSE=ImageQuilting(in_imgS , in_imgE , in_overlap , path);
23 %
24 imgWnr=rot90(imgWN);
25 imgSEr=rot90(imgSE);
26 overWN=imgWnr(:, size(imgWnr,2)-in_overlap+1:size(imgWnr
    ,2));
27 overSE=imgSEr(:, 1:in_overlap);
28 %e=(double(overWN)-double(overSE)).^2;
29 e=ECreate(overWN , overSE);
30 e=ErrorMask(e);
31 e=flipud(e); % necessary
32 e=ErrorMask(e);
33 e=flipud(e); % necessary
34 E=CumulativeError(e);
35 [path , error]=MinErrorPath(E);
36 imgWNSE=ImageQuilting(imgWnr , imgSEr , in_overlap , path);
37 %
38 out_img=ImageRotationCrop( rot90(imgWNSE,-1) ,
    in_overlap , 'bilinear');
39 %
40 end

```

Algoritmus 4: Procedura *ECreate*

```

1 function [ out_e ] = ECreate( in_imgA , in_imgB )
2 %ECREATE creates modified error matrix
3 nr=size(in_imgA , 1);

```

```

4 nc=size(in_imgA,2);
5 out_e=zeros(nr,nc);
6 for i=1:nr
7     for j=1:nc
8         if (in_imgA(i,j)==0)&&(in_imgB(i,j)==0)
9             out_e(i,j)=0;
10        elseif (in_imgA(i,j)==1)&&(in_imgB(i,j)==1)
11            out_e(i,j)=1;
12        elseif in_imgA(i,j)~=in_imgB(i,j)
13            out_e(i,j)=2;
14        end;
15    end;
16 end;
17 end

```

Algoritmus 5: Procedura *CumulativeError*

```

1 function [ out_E ] = CumulativeError( in_e )
2 %CUMULATIVEERROR calculates cumulative error introduces
   by Efros and Freeman
3 out_E=zeros(size(in_e));
4 % begining in the top left corner
5 out_E(1,:)=in_e(1,:);
6 for i=2:size(in_e,1)
7     for j=1:size(in_e,2)
8         out_E(i,j)=in_e(i,j)+min( out_E(i-1, max(1,j-1)
           :min(j+1,size(in_e,2))) );
9     end;
10 end;
11 end

```

Algoritmus 6: Procedura *ErrorMask*

```

1 function [ out_e ] = ErrorMask( in_e )

```



```

2 %ERRORMASK adds error penalties to restrict a position
   of a image quilt
3 n=size(in_e,2);
4 out_e=in_e;
5 out_e(1:ceil(n/2),:)=10^20;
6 out_e(1:ceil(n/2),ceil(n/2))=10^10; % restricts final
   cut to the centre
7 % wing-shaped penalties
8 for i=1:(n-ceil(n/2))
9     if (floor(n/2)-i)~=0
10         out_e(ceil(n/2)+i,1:(floor(n/2)-i))=10^20;
11     end;
12     out_e(ceil(n/2)+i,(ceil(n/2)+i):n)=10^20;
13 end;
14 end

```

Algorithmus 7: Procedura *MinErrorPath*

```

1 function [ out_pth, out_err ] = MinErrorPath( in_E )
2 %MINERRORPATH searches for the minimal error path
   within a input matrix in_E
3 % out_pth is a vector of points on the path with the
   lowest cumulative error
4 % out_err overall path error
5 out_pth=zeros(size(in_E,1),1);
6 %out_err=0;
7 for i=1:size(in_E,1)
8     % j is an inverse index since the error is
   cumulated from the top
9     j=size(in_E,1)+1-i;
10    if i==1
11        % searches for the minimum within the last row
12        [ derr, out_pth(j,1)]=min(in_E(j,:));

```

```

13         %out_err=out_err+derr;
14         out_err=derr;
15     else
16         % searches for the minimum within a one-pixel
           reach form the lower row minimum
17         [derr,dpth]=min( in_E(j, max( 1, out_pth(j+1,1)
           -1 ) : min( out_pth(j+1,1)+1 , size(in_E,2)
           ) ) );
18         %out_err=out_err+derr;
19         out_pth(j,1)=max(0,out_pth(j+1,1)-2)+dpth;
20     end;
21 end;
22 end

```

Algoritmus 8: Procedura *ImageQuilting*

```

1 function [ out_AB ] = ImageQuilting( in_A, in_B,
   in_over, in_pth )
2 %QUILT stitches two images together with in_over
   overlap according to cutting path in_pth
3 % input images must be the same size => size(in_A) =
   size(in_B)
4 % points on the cutting path are taken form the left
   image in_A
5 n1=size(in_A,1);
6 n2=size(in_A,2); %=size(in_B,2)
7 out_AB=(zeros( n1 , n2+n2-in_over ));
8 for i=1:n1
9     out_AB(i,:)=[ in_A( i,1:n2-in_over+in_pth(i) ) ,
   in_B(i,in_pth(i)+1:n2) ];
10 end;
11 end

```

Algoritmus 9: Procedura *ImageRotationCrop*

```

1 function [ out_image ] = ImageRotationCrop( in_image ,
      in_over , in_method )
2 %IMAGEROTATIONCROP rotates and crops input image after
      image quilting
3 % in_image ... input image in binary format
4 % in_over ... size of overlapping area used for image
      quilting
5 % in_method ... method used for image interpolation
6 %          either 'nearest' or {'bilinear'}
7 Xa=in_over/2;
8 Ya=size(in_image,1)/2;
9 Xb=Ya;
10 Yb=Xa;
11 CornersDistance=sqrt((Xb-Xa)^2+(Yb-Ya)^2);
12 ImageOutSize=ceil(CornersDistance);
13 Scale=CornersDistance/ImageOutSize;
14 out_image=im2bw(zeros(ImageOutSize));
15 Angle=-pi/4;
16 TransformMatrix=[Scale*cos(Angle), Scale*sin(Angle), 0;
17 -Scale*sin(Angle), Scale*cos(Angle), 0;
18 Xa, Ya, 1];
19 %
20 for i=1:ImageOutSize
21     for j=1:ImageOutSize
22         Coordinates=[i, j, 1]*TransformMatrix;
23         X=Coordinates(1);
24         Y=Coordinates(2);
25         if strcmpi('nearest', in_method)
26             out_image(i, j)=in_image( round(X) , round(Y
      ) );

```

```

27         elseif strcmpi( 'bilinear' ,in_method)
28             P1=in_image( floor(X) ,floor(Y) );
29             P2=in_image( floor(X) ,ceil(Y) );
30             P3=in_image( ceil(X) ,floor(Y) );
31             P4=in_image( ceil(X) ,ceil(Y) );
32             P5=P1+(P3-P1)*(X-floor(X) );
33             P6=P2+(P4-P2)*(X-floor(X) );
34             out_image(i ,j)=round(P5+(P6-P5)*(Y-floor(Y)
                ));
35         end;
36     end;
37 end;
38 end

```

Algoritmus 10: Procedura *TwoPointProbability*

```

1 function [ out_graph ] = TwoPointProbability( in_img ,
        in_method , in_nr )
2 %TWOPOINTPROBABILITY Summary of this function goes here
3 A=in_img;
4 [ nRow , nCol]=size(A);
5 phase=1; % defining phase
6 A=im2bw(A); % conversion to a binary image
7 %
8 if strcmpi(in_method , 'direct') % Direct computation
9     tic;
10    % generating vector list
11    vL=[];
12    for i=0:(nRow-1) % n possibilities
13        for j=(-nCol+1):(nCol-1) % 2n-1
            possibilities
14            vL=[vL ; i , j ];
15    end;

```

```

16     end;
17     % direct computation of two-point probability
        function
18     B=[A,A,A;A,A,A];
19     tpp=zeros(nRow,2*nCol-1);
20     for i=1:nRow
21         for j=(nCol+1):(nCol*2)
22             if B(i,j)==phase
23                 for k=1:size(vL,1)
24                     if (B(i+vL(k,1),j+vL(k,2))==phase)
25                         tpp(1+vL(k,1),nCol+vL(k,2))=tpp
26                             (1+vL(k,1),nCol+vL(k,2))+1;
27                     end;
28                 end;
29             end;
30     end;
31     % post-processing
32     out_graph=[rot90(tpp(2:size(tpp,1),:),2);tpp]./(
33         nRow*nCol);
34     %
35     elseif strcmpi(in_method,'MonteCarlo') % MonteCarlo
        method
36         %tic;
37         % generating vector list
38         vL=[];
39         for i=0:(nRow-1) % n possibilities
40             for j=(-nCol+1):(nCol-1) % 2n-1
                possibilities
41                 vL=[vL;i,j];

```

```

42         end;
43     end;
44     % Monte-Carlo
45     B=[A,A,A;A,A,A];
46     tpp=zeros(nRow,2*nCol-1);
47     for i=1:size(vL,1)
48         r=0; % number of right hits
49         for j=1:round(in_nr*(nCol*nRow))
50             x=ceil(nRow*rand(1));
51             y=nCol+ceil(nCol*rand(1));
52             if (B(x,y)==phase)&& (B(x+vL(i,1),y+vL(i,2))
53                 ==phase)
54                 r=r+1;
55             end;
56         end;
57         tpp(1+vL(i,1), nCol+vL(i,2))=r/round(in_nr*(
58             nCol*nRow));
59     end;
60     % post-processing
61     out_graph=[rot90(tpp(2:size(tpp,1),:),2); tpp];
62     %toc;
63     %
64     else % Discrete Fourier Transform
65         %tic;
66         C=fft2(A);
67         q1=ifft2(C.*conj(C))./(nRow*nCol);
68         D=fft2(rot90(A));
69         q2=rot90((ifft2(D.*conj(D))./(nRow*nCol)),-1)
70             ;
71         q3=rot90(q1,2);
72         q4=rot90(q2,2);

```

```
70     out_graph=[ q3(1:nRow-1,1:nCol-1), q4(1:nRow-1,:);
71                q2(:,1:nCol-1), q1(:,,:) ] ;
72     %t o c
73 end ;
74
75 end
```

B Algoritmus otáčení a ořezávání dlaždic

Aby bylo možné Wangovy dlaždice konstruovat dle postupu popsaného v kapitole 3.1, je nutné výsledek vzniklý spojením vzorků W,N,E a S otočit o 45° a oříznout. Tohoto nebylo možné dosáhnout pomocí příkazů *imrot* a *imcrop* v prostředí Matlab, jelikož neumožňují explicitně určit, jakou část otočeného WNES obrázku je nutné vyříznout. Z tohoto důvodu byl implementován následující algoritmus.

Ze známých souřadnic bodů A a B (viz obr. 16), tj. $A = [o/2, a/2]$ a $B = [a/2, o/2]$, je stanovena velikost výsledné dlaždice b jako na přirozené číslo zaokrouhlená norma vektoru $\|AB\|$. Následně je vytvořena prázdná bitmapa o velikosti $b \times b$. Postupně je hodnota každého bodu výsledné matice stanovena interpolací z původní matice.

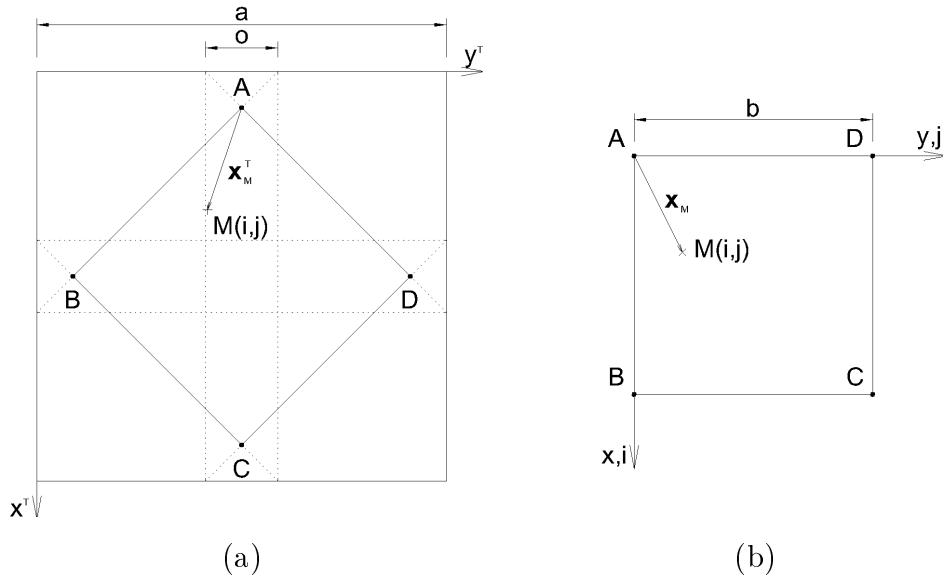
Pro každý bod M výsledné matice, tj. prvek matice $ImageOut(i,j)$, je stanoven polohový vektor $\mathbf{x}_M = (i - \frac{1}{2}, j - \frac{1}{2}, 1)$. Hodnota $\frac{1}{2}$ v definici vektoru \mathbf{x}_M je odečítána z důvodu rozdílné interpretace diskretizace v prostředí Matlab a z pohledu teorie počítačové grafiky. Při načtení obrázku do prostředí Matlab jsou uloženy informace o hodnotách pixelů do prvků matice, jejichž indexy jsou celočíselné, nicméně diskrétní hodnota prvku matice $ImageOut(i,j)$ přísluší čtvercovému pixelu o hraně jednotkové délky se středem v bodě $\mathbf{x}_M = (i - \frac{1}{2}, j - \frac{1}{2})$. Tento vektor je navíc rozšířen na $\mathbf{x}_M = (i - \frac{1}{2}, j - \frac{1}{2}, 1)$ s ohledem na tvar transformační matice. Vektor \mathbf{x}_M je transformován dle výrazu [13]

$$\mathbf{x}^T = \mathbf{x}\mathbf{A}, \quad (22)$$

kde \mathbf{A} je transformační matice definovaná v [13] jako

$$\mathbf{A} = \begin{pmatrix} m_x \cos(\alpha) & m_y \sin(\alpha) & 0 \\ -m_y \sin(\alpha) & m_x \cos(\alpha) & 0 \\ X_A & Y_A & 1 \end{pmatrix}, \quad (23)$$

m_x a m_y zastupuje koeficient změny měřítka, α je úhel rotace souřadnic a hodnoty $[X_A, Y_A]$ jsou souřadnice udávající translaci systému. V případě

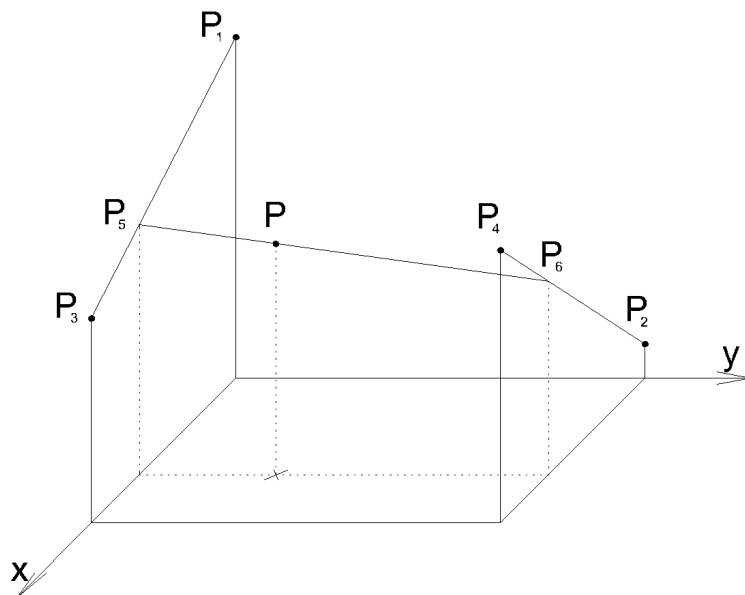


Obrázek 16: Schéma otočení a ořezu (a) spojených vzorků W,N,S,E a (b) výsledné dlaždice

Wangových dlaždic, které jsou vždy čtvercové, platí $m_x = m_y = \frac{\|AB\|}{b}$, $\alpha = 45^\circ$ a $[X_A, Y_A] = [o/2, a/2]$.

Hodnota v bodě $ImageOut(i,j)$ je stanovena interpolací z okolí bodu $ImageIn(\mathbf{x}^T(1) + \frac{1}{2}, \mathbf{x}^T(2) + \frac{1}{2})$ umístěného v původním vzorku buď výběrem hodnoty dle nejbližšího bodu, anebo stanovením hodnoty na základě bilineární transformace ze čtyřech okolních bodů, viz obr. 17.

Na závěr je vhodné poznamenat, že vlivem interpolace dochází k určité deformaci analytických tvarů, zejména pro vstupní mikrostruktury v nízkém rozlišení (struktura s disky). V případě reálných mikrostruktur dochází k určité deformaci tvaru inkluzí již při získávání záznamu (skenováním, CT atd.), takže malé odchylky vzniklé transformací lze tolerovat.



Obrázek 17: Schéma bilineární transformace

C Diskrétní Fourierova transformace

Fourierova transformace (FT) je rozšířením interpolace funkce rozvojem do trigonometrických řad, kdy sumu konečného počtu členů Fourierovy řady nahrazuje integrální součet. Tato transformace převádí časově závislé funkce $f(t)$ na funkce frekvence $\tilde{f}(\omega)$. Jednou z výhod této transformace je zvýraznění některých vlastností původní funkce $f(t)$ ve Fourierových obrazech [14]. FT umožňuje efektivní přenos a ukládání především multimediálních dat (např. komprimace a přenos hlasového záznamu, JPEG komprese obrázků založená na kosinové transformaci, atd. [3, 14]). V neposlední řadě jde o výrazné zjednodušení některých matematických operací, jako je konvoluce nebo korelace dvou funkcí.

Pro reálnou spojitou funkci $f(t)$ je Fourierova transformace definována vztahem

$$\mathcal{F}(f(t)) = \tilde{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt, \quad (24)$$

kde i je imaginární jednotka. V případě inverzní transformace platí

$$f(t) = \mathcal{F}^{-1}(\tilde{f}(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{f}(\omega)e^{i\omega t} d\omega. \quad (25)$$

S ohledem k diskusi v kapitole 1.2, mikrografy lze chápat jako diskrétní popis spojitého média a v tomto případě je nutné nahradit spojitou Fourierovu transformaci její diskrétní formou. Diskrétní Fourierova transformace (DFT) pro N diskrétních hodnot funkce $f(k)$ je definována jako [15]

$$\mathcal{F}_{DFT}(f(k)) = \tilde{f}(n) = \sum_{k=0}^{N-1} f(k)e^{-2\pi i n k / N}. \quad (26)$$

Inverzní DFT lze vyjádřit výrazem

$$f(k) = \mathcal{F}_{DFT}^{-1}(\tilde{f}(n)) = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{f}(n)e^{2\pi i n k / N}. \quad (27)$$

Výše uvedené vztahy platí pro jednodimenzionální problém, lze je však snadno rozšířit pro funkce více proměnných [14]. V rámci této bakalářské práce byla diskrétní Fourierova transformace v rovnicích (17,18) z kapitoly 1

vyhodnocována v prostředí Matlab pomocí příkazů *fft2* a *ifft2*. Ty pro výpočet DFT využívají algoritmu rychlé Fourierovy transformace (Fast Fourier Transform) [16].