VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING INSTITUTE OF MATHEMATICS

APLIKACE ŘÍDKÝCH REPREZENTACÍ DAT

DIPLOMOVÁ PRÁCE MASTER'S THESIS

AUTOR PRÁCE

Bc. BARBORA NAVRÁTILOVÁ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING INSTITUTE OF MATHEMATICS

APLIKACE ŘÍDKÝCH REPREZENTACÍ DAT

APPLICATIONS OF SPARSE DATA REPRESENTATIONS

DIPLOMOVÁ PRÁCE MASTER'S THESIS

AUTOR PRÁCE

Bc. BARBORA NAVRÁTILOVÁ

VEDOUCÍ PRÁCE SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Barbora Navrátilová

který/která studuje v magisterském navazujícím studijním programu

obor: Matematické inženýrství (3901T021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Aplikace řídkých reprezentací dat

v anglickém jazyce:

Applications of sparse data representations

Stručná charakteristika problematiky úkolu:

Využívání tzv. řídké reprezentace dat, tzn. zjištění, že data se často skládají z relativně malého počtu stavebních bloků vůči celkové délce, je dnes vysoce aktuální téma. Metodami lineární algebry a optimalizace lze nejen řídké vyjádření získat, ale také vyžít k nejrůznějším praktickým aplikacím. Smyslem práce je nastudovat základní teorii řídké reprezentace dat, porozumět principům současných algoritmů pro hledání řídkých řešení a empiricky otestovat účinnost vybraných metod na reálných datech (jedno- a dvoudimenzionální signály).

Cíle diplomové práce:

Nastudovat základní principy a algoritmy hledání tzv. řídkých reprezentací.

Udělat si přehled v aplikačních oblastech a pro praktické ověření metod na simulovaných i reálných datech zvolit např. odstraňování šumu, dekvantizaci, morfologickou analýzu komponent. Shrnout a interpretovat výsledky.

ABSTRAKT

Cílem této práce je přiblížení problematiky řídké reprezentace dat a ukázka praktických aplikací v oblasti zpracování řídkého signálu. K řešení bylo použito metod z oblasti konvexní optimalizace. Byla řešena úloha odstanění šumu, dekvantizace a dekompozice řídkého signálu. Úlohy byly implementovány v prostředí MATLAB. K jednotlivým úlohám byly vytvořeny programy pro jednorozměrný a dvourozměrný signál.

KLÍČOVÁ SLOVA

Řídká reprezentace, odšumování, Dantzig selektor, dekvantizace, Morfologická analýza komponent

ABSTRACT

The goal of this thesis is to demonstrate practical application of sparse data representation in the processing of sparse signals. For solving several example problems denoising, dequantization, and sparse signal decomposition - convex optimization was used. The solutions were implemented in the Matlab environment. For each of the problems, there are two solutions - one for one-dimensional, and one for two-dimensional signal.

KEYWORDS

Sparse representation, Denoising, Dantzig Selector, Dequantization, Morphological Component Analysis

NAVRÁTILOVÁ, Barbora APLIKACE ŘÍDKÝCH REPREZENTACÍ DAT: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav matematiky, 2014. 72 s. Vedoucí práce byl Mgr. Pavel Rajmic, Ph.D.

Prohlašuji, že jsem diplomovou práci *Aplikace řídkých reprezentací dat* vypracovala samostatně s použitím odborné literatury a pramenů, uvedených na seznamu, jenž je součástí této práce.

 $30.5.\ 2014$

Barbora Navrátilová

Děkuji panu Mgr. Pavlu Rajmicovi, Ph.D. za rady, věnovaný čas a odborné vedení při tvorbě této diplomové práce.

Barbora Navrátilová



Faculty of Electrical Engineering and Communication Brno University of Technology Purkynova 118, CZ-61200 Brno Czech Republic http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

(podpis autora)





EVROPSKÁ UNIE EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Ú	vod		9
1	Říd	ká reprezentace signálů	11
	1.1	Základní pojmy	11
2	Říd	ká rešení soustav lineárních	
	rovi	nic	13
	2.1	Formulace základní úlohy	13
	2.2	Postačující podmínky jednoznačnosti	13
	2.3	Možnosti volby slovníku	15
		2.3.1 Pevné slovníky	15
		2.3.2 Trénované slovníky	18
	2.4	Aproximativní metody	19
		2.4.1 Relaxační algoritmy	19
3	Pot	lačování šumu v řídkém signálu	23
	3.1	Zobecnění úlohy	23
		3.1.1 Algoritmus Forward-Backward	25
		3.1.2 Algoritmus Douglas-Rachford	27
4	Dar	ntzig Selektor	29
	4.1	Ekvivalence řešení BPDN a DS	30
	4.2	Podmínky řešitelnosti	30
	4.3	Variace a řešiče DS	31
5	Dek	vantizace řídkého signálu	33
	5.1	Formulace úlohy dekvantizace	34
6	Mo	rfologická analýza komponent	37
	6.1	Globální MCA	38
	6.2	Lokální MCA	38
7	Rea	lizace algoritmů v prostředí MATLAB	41
	7.1	1D Dekvantizace	41
	7.2	2D Dekvantizace	46
	7.3	1D a 2D Dekompozice	51
	7.4	1D a 2D Odšumování signálu	53

Li	Literatura 59								
\mathbf{Se}	znan	n přílo	h	61					
\mathbf{A}	Prv	ní příl	oha	63					
	A.1	Přilože	ené programy – popis funkcí	63					
		A.1.1	2D úlohy	64					
		A.1.2	Možnosti vyhození chyby	64					
		A.1.3	Ukázka uživatelského prostředí	64					
в	Dru	há pří	loha	67					
	B.1	Trénov	vané slovníky	67					
\mathbf{Se}	Seznam symbolů, veličin a zkratek 6								

ÚVOD

Diplomová práce Aplikace řídkých reprezentací dat pojednává o problematice řídké reprezentace dat. Tato problematika je v posledních letech stále populárnější. Hlavní myšlenkou je vyjádřit signál v co nejmenší možné reprezentaci, tedy hledáme takový systém – bázi, ve kterém bude mít signál co největší počet nulových složek. Tuto úlohu si lze rovněž představit jako úlohu řešení nedourčeného systému rovnic, kde hledané řešení je to s největším počtem nul. Jedná se tedy o optimalizační úlohu. Takové vyjádření signálu umožní jeho snadnější interpretaci a také se pozitně projeví i v mnoha aplikacích.

V úvodu této práce zavádíme pojem řídkosti a seznamujeme se s podmínkami, kdy je úlohu najít "řídké řešení" možno řešit pomocí metod konvexní optimalizace. Tyto metody skýtají široké možnosti řešení minimalizačních úloh.

Další kapitoly již pokládají teoretický základ pro konkrétní aplikace – odstranění šumu, dekvantizace a dekompozice. Každá uvedená úloha byla řešena jinou metodou. Problematiku odstranění šumu řešíme přes proximity operátor Douglas-Rachford algoritmem. Dále řešíme úlohu dekvantizace, tj. řešení možné rekonstrukce kvantovaného signálu zpět na původní signál. Využíváme myšlenky z Dantzig Selektoru. Poslední aplikací je dekompozice signálu, tedy úloha rozkladu signálu na jednotlivé komponenty. Tato úloha je řešena přes Forward-Backward algoritmus.

Všechny uvedené aplikace jsou zprogramovány pro jenorozměrný i dvourozměrný signál v softwaru MATLAB v uživatelském prostředí GUI. Zdrojové kódy jsou přiloženy k práci a popis jednotlivých funkcí je možné nalézt v příloze.

1 ŘÍDKÁ REPREZENTACE SIGNÁLŮ

Zpracování signálů hraje významnou roli v různých technických odvětvích i v současné vědě. V několika posledních letech se možnost vyjádření signálu pomocí nedourčeného systému lineárních rovnic představuje jako vhodný prostředek reprezentace signálu. Řešení nedourčeného systému lineárních rovnic je nekonečně mnoho. Pro nás jsou nejvhodnejší ta, která mají současně co nejvíce neznámých rovno nule – řídká řešení. Tato volba nám poskytne schopnost snadnější interpetace dat, rychlejší komprese a je numericky stabilnější. Problematika řídké reprezentace zasahuje do mnoha oblastí matematiky jako je lineární algebra, funkcionální analýza, konvexní optimalizace aj.

1.1 Základní pojmy

Definice 1.1. [2] Nosičem vektoru **x** rozumíme množinu jeho indexů, v nichž má vektor nenulové hodnoty. Tuto množinu značíme supp (**x**). Tedy supp (**x**) = $\{i \mid x_i \neq 0\}$.

Definice 1.2. [2] Nechť $\mathbf{x} \in \mathbb{C}^n$. Pak ℓ_p -norma vektoru \mathbf{x} je dána jako

$$\|\mathbf{x}\|_{p} = |\operatorname{supp}(\mathbf{x})| \quad \text{pro} \qquad p = 0$$
$$\|\mathbf{x}\|_{p} = \sum_{i=1}^{n} |x_{i}|^{p} \quad \text{pro} \quad 0
$$\|\mathbf{x}\|_{p} = \left(\sum_{i=1}^{n} |x_{i}|^{p}\right)^{\frac{1}{p}} \quad \text{pro} \quad 1 \le p < \infty$$
$$\|\mathbf{x}\|_{p} = \max_{i} |x_{i}| \quad \text{pro} \qquad p = \infty$$$$

Poznámka 1.3. O normu se ve striktním slova smyslu jedná pouze v případě $1 \le p \le \infty$. Pro $0 \le p < 1$ není splněna podmínka trojúhelníkové nerovnosti, nejedná se tedy o konvexní záležitost. V následujícím textu budeme ovšem pro zjednodušení hovořit ve všech případech tj. $0 \le p \le \infty$ o l_p normě.

Definice 1.4. [3] Necht $\mathbf{A} \in \mathbb{C}^{n \times m}$. Pak Frobeniova norma matice \mathbf{A} je dána

$$\|\mathbf{A}\|_{F} = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^{2}\right)^{\frac{1}{2}}.$$
(1.2)

Definice 1.5. [2] Necht $\mathbf{A} \in \mathbb{C}^{m \times n}$. Pak norma $\|\cdot\|_{2 \to 2}$ matice \mathbf{A} je dána

$$\|A\|_{2\to 2} = \max_{\mathbf{x}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$
 (1.3)

Definice 1.6. [2] Vektor $\mathbf{x} \in \mathbb{C}^n$ nazveme k-*řídkým*, pokud platí

$$\|\mathbf{x}\|_0 \le k. \tag{1.4}$$

Relativní řídkostí vektoru **x** délky n rozumíme výraz $\frac{k}{n}$. Množinu všech k-řídkých vektorů délky n označme Σ_k . Tedy $\Sigma_k = \{ \mathbf{x} \in \mathbb{C}^n \mid ||\mathbf{x}||_0 \leq k \}.$

Definice 1.7. [2] Necht $\mathbf{x} \in \mathbb{C}^n$. *Chyba nejlepší aproximace* vektoru \mathbf{x} k-řídkým vektorem v l_p normě je dána

$$\sigma_k \left(\mathbf{x} \right)_p = \sigma_k^n \left(\mathbf{x} \right)_p = \inf_{\mathbf{z} \in \Sigma_k^n} \left\| \mathbf{x} - \mathbf{z} \right\|_p.$$
(1.5)

Poznámka 1.8. Jak si lze povšimnout, chyba je závislá vždy na konkrétním \mathbf{x} , jedná se tedy o *adaptivní* problém.

Definice 1.9. [3] Necht $\mathbf{A} \in \mathbb{C}^{n \times m}$, $\mathbf{A} = [a_{i,j}]$, $i = 1, \dots, n, j = 1, \dots, m$. Matici hermitovsky sdruženou k matici \mathbf{A} rozumíme matici $\mathbf{A}^* \in \mathbb{C}^{m \times n}$ s prvky $\bar{a}_{j,i}$ na pozicích i, j pro $i = 1, \dots, m, j = 1, \dots, n$, kde $\bar{a}_{j,i}$ označuje číslo komplexně sdružené k číslu $a_{j,i}$.

Definice 1.10. [3] Řekneme, že čtvercová matice A je unitární, jestliže platí

$$\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^* = \mathbf{I},\tag{1.6}$$

kde I je jednotková matice a A^* je matice hermitovsky sdružená k matici A.

2 ŘÍDKÁ REŠENÍ SOUSTAV LINEÁRNÍCH ROVNIC

2.1 Formulace základní úlohy

Jak bylo již řečeno, řídkou reprezentaci signálu budeme chápat jako nedourčený systém soustavy lineárních rovnic, tedy $\mathbf{A}\mathbf{x} = \mathbf{y}$, kde matice \mathbf{A} je matice koeficientů, \mathbf{y} je známý vektor (pozorování, měření, signál). Úkolem je nalézt takové \mathbf{x} , které bude co nejřidší. Tedy [1]

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{0} \quad \text{vzhledem k} \quad \mathbf{A}\mathbf{x} = \mathbf{y}, \tag{P0}$$

kde $\mathbf{y} \in \mathbb{C}^m$. Předpokládáme matici $\mathbf{A} \in \mathbb{C}^{m \times n}$ plné hodnosti a budeme ji nazývat slovník, její sloupce nazýváme atomy. Všechna \mathbf{x} vyhovující rovnici $\mathbf{A}\mathbf{x} = \mathbf{y}$ budeme nazývat přípustná řešení.

V případě šumu uvažujeme úlohu [1]

$$\min_{\mathbf{y}} \|x\|_0 \quad \text{vzhledem k} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \le \epsilon, \tag{P0}\epsilon$$

kde ϵ je uvažovaná odchylka od přesného řešení.

2.2 Postačující podmínky jednoznačnosti

Pro řešení otázky existence a jednožnačnosti řešení si zavedeme následující pojem.

Definice 2.1. [2] Číslo spark(\mathbf{A}) definujeme jako nejmenší počet sloupců matice \mathbf{A} , které jsou lineárně závislé. Formálně

$$\operatorname{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \ker \mathbf{A}, \mathbf{z} \neq 0} \|\mathbf{z}\|_0.$$
(2.1)

Poznámka 2.2. Hodnotu spark(\mathbf{A}) tedy můžeme chápat jako protipól hodnosti matice rank(\mathbf{A}) (vyjadřuje nejmenší počet sloupců matice právě lineárně nezávislých).

Pro nenulovou matici $\mathbf{A} \in \mathbb{C}^{m \times n}$, kde m < n, může nabývat hodnot spark $(\mathbf{A}) = \{2, \ldots, m+1\}$, přičemž hodnoty 2 je dosaženo v případě, pokud je jeden sloupec přímo násobkem jiného.

Tvrzení 2.3. [1] Pokud má soustava Ax = b řešení x splňující

$$\|\mathbf{x}\|_0 < \frac{\operatorname{spark}(\mathbf{A})}{2},\tag{2.2}$$

pak \mathbf{x} je nutně nejřidší možné řešení a žádné jiné řešení se stejnou řídkostí neexistuje.

 $d \hat{u} kaz: [1]$

Jedná se ale pouze o postačující podmínku. Výpočtová náročnost pro nalezení hodnoty spark je však srovnatelná s přímým řešením základní úlohy. Na základě této skutečnosti se snažíme stanovit přijatelnější podmínky pro ověření jednoznačnosti řešení.

Definice 2.4. Vzájemná koherence matice \mathbf{A} je definována jako nejvetší absolutní normalizovaný skalární součin dvou různých sloupců matice \mathbf{A} ,

$$\mu(\mathbf{A}) = \max_{1 \le j, k \le N, j \ne k} \frac{\left|\mathbf{a}_{j}^{T} \mathbf{a}_{k}\right|}{\left\|\mathbf{a}_{j}\right\|_{2} \cdot \left\|\mathbf{a}_{k}\right\|_{2}},$$
(2.3)

kde \mathbf{a}_j označuje *j*-tý sloupec matice \mathbf{A}

Vzájemná koherence nám jistým způsobem určuje "míru lineární závislosti", přičemž v našem případě bude vždy nenulová. S omezením

$$\sqrt{\frac{N-m}{m(N-1)}} \le \mu(\mathbf{A}) \le 1.$$

Tvrzení 2.5. [1] Pro libovolnou matici A platí

$$\operatorname{spark}(\mathbf{A}) \ge 1 + \frac{1}{\mu(\mathbf{A})}.$$
 (2.4)

důkaz:[1]

Na základě tohoto tvrzení je možné stanovit dolní mez hodnoty spark. Jde o výpočtově výrazně méně náročný problém. Jednoznačnost řešení tedy bude určena následujícím tvrzením.

Tvrzení 2.6. [1] Pokud má soustava Ax = b řešení x splňující

$$\|\mathbf{x}\|_{0} < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})} \right),$$
 (2.5)

pak \mathbf{x} je nutně nejřidší možné a je jediné takové. Navíc toto řešení lze dosáhnout ℓ_1 -minimalizací.

dukaz:[1]

V našem případě je přirozené hledat maximálně nekorehentní slovníky.

2.3 Možnosti volby slovníku

Významnou roli v definici hraje především slovník – matice \mathbf{A} . Nabízí se otázka, jak vhodně zvolit \mathbf{A} , aby řešení existovalo a bylo co nejřidší. Budeme rozlišovat dva základní přístupy:

- slovník A je předem stanovený slovník je zde často označován jako transformace, umožňuje rychlé zpracování o nízké složitosti, je však omezen na určité druhy signálu
- slovník A je vytvořen signálu na míru založeno na trénovací databázi, lze jej užít na libovolnou skupinu signálů, ale za cenu mnohem větší výpočetní náročnosti

2.3.1 Pevné slovníky

Jednou z možností sestavení slovníku \mathbf{A} je spojení dvou ortonormálních bází např. Diracova báze v časové oblasti a matice Fourierovy transformace tj. $\mathbf{A} = [\mathbf{I} \mathbf{F}]$. Další z variant je diskrétní kosinová transformace (narozdíl od Fourierovy transformace má reálné koeficienty), diskrétní vlnková transformace, využití framů nebo matice Gaussova či Bernoulliho typu.

Diskrétní kosinová transformace – DCT

Uvažujme signál Y jako původní signál, signál X jako signál transformovaný.

 Jednorozměrná diskrétní kosinová transformace Koeficienty DCT jsou dány jako:

$$\mathbf{X}_{i} = \alpha_{i} \sum_{j=0}^{n-1} \mathbf{Y}_{j} \cos\left(\frac{\pi \left(2j+1\right)i}{2n}\right), \qquad (2.6)$$

kde

$$\alpha_i = \begin{cases} \sqrt{\frac{1}{n}}, & \text{pro} \quad i = 0\\ \sqrt{\frac{2}{n}}, & \text{pro} \quad 1 \le i \le n - 1 \end{cases}$$

• Dvourozměrná diskrétní kosinová transformace

Vícerozměrnou DCT lze vypočítat dvěma způsoby. Buď jako sérii jednorozměrných transformací postupně v každém rozměru, v našem případě transformace nejprve po řádcích a poté po sloupcích - pořadí lze zaměnit, a nebo přímo jsou koeficienty DCT dány jako:

$$\mathbf{X}_{i_1,i_2} = \alpha_{i_1} \alpha_{i_2} \sum_{j_1=0}^{n-1} \sum_{j_2=0}^{m-1} \mathbf{Y}_{j_1,j_2} \cos\left(\frac{\pi \left(2j_1+1\right) i_1}{2n}\right) \cos\left(\frac{\pi \left(2j_2+1\right) i_2}{2m}\right), \quad (2.7)$$

kde

$$\alpha_{i_1} = \begin{cases} \sqrt{\frac{1}{n}}, & \text{pro} \quad i = 0, \\ \sqrt{\frac{2}{n}}, & \text{pro} \quad 1 \le i \le n - 1, \end{cases}$$
$$\alpha_{i_2} = \begin{cases} \sqrt{\frac{1}{m}}, & \text{pro} \quad i = 0, \\ \sqrt{\frac{2}{m}}, & \text{pro} \quad 1 \le i \le m - 1. \end{cases}$$

Další možností výpočtu DCT je pomocí matice \mathbf{T} diskrétní kosinové transformace. Prvky matice jsou dány:

$$\mathbf{T}_{ij} = \begin{cases} \frac{1}{\sqrt{n}}, & \text{pro} \quad i = 0, \quad 0 \le j \le n - 1\\ \sqrt{\frac{2}{n}} \cos \frac{\pi(2j+1)i}{2n}, & \text{pro} \quad 1 \le i \le n - 1, \quad 0 \le j \le n - 1. \end{cases}$$

Pak transformovaný signál můžeme zapsat jako

 $\mathbf{X} = \mathbf{T} \mathbf{Y} \mathbf{T}^{\mathrm{T}}.$

Diskrétní waveletová transformace – DWT

V diskrétním případě je teorie waveletů založena na jednoduché myšlence rozkladu vektoru dat vzhledem k soustavě vektorů. Tato soustava je generována jednou nebo několika funkcemi (tzv. wavelety) pomocí jejich dilatace a translace. Hlavní roli v DWT hraje pojem diskrétní lineární konvoluce a waveletové filtry. Diskrétní lineární konvolucí dvou signálů $\mathbf{x} = [x_1, x_2, \ldots, x_s]$ a $\mathbf{h} = [h_1, h_2, \ldots, h_m]$ je vektor \mathbf{y} :

$$y_n = (x * h)_n = \sum_{k=0}^{m-1} x_{n-k} h_{1+k}, \qquad (2.8)$$

kde $n \in \{1, \ldots, s + m - 1\}$. Problémy spojené s konečnou délkou signálu, která nám způsobí okrajovými jevy konvoluce, budeme řešit periodickým prodloužením okrajů.

Označme **h** rozklad pomocí filtru typu dolní propust, **g** rozklad pomocí filtru typu horní propust, $\tilde{\mathbf{h}}$ a $\tilde{\mathbf{g}}$ rekonstrukce pomocí těchto filtrů.

DWT signálu **x** můžeme získat také skrze násobení matic. Uvažujme signál **x** = $[x_1, x_2, \ldots, x_s]$ o délce $s = 2^J$, kde J označuje stupeň dekompozice. Tento signál je možné reprezentovat jako lineární kombinaci vlnkových vektorů uspořádaných ve sloupcích matice **W**. Rozměr matice **W** je $s \times s$. Koeficienty této kombinace jsou umístěny ve vektoru **y**. Pak **x** = **Wy**. Matici DWT pro hloubku dekompozice J = 1 můžeme zapsat jako

$$\mathbf{W}^{-1} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{G}_1 \end{bmatrix},\tag{2.9}$$

kde řádky \mathbf{H}_1 a \mathbf{G}_1 jsou tvořeny pomocí vektorů $\mathbf{h} = [h_1, \dots, h_m]$ a $\mathbf{g} = [g_1, \dots, g_m]$ tak, že následující řádek je posunut o dvě pozice doprava. Obecná konstrukce matice W pro různou hloubku dekompozice je uvedena v [4].

Ukažme na příkladu, jak může taková matice vypadat. Pokud budeme uvažovat ortogonální filtry $\mathbf{h} = [h_1, h_2, h_3, h_4]$ a $\mathbf{g} = [g_1, g_2, g_3, g_4]$ a délku signálu s = 8. Obdržíme matici:

$$\mathbf{W}^{-1} = \begin{pmatrix} h_3 & h_2 & h_1 & 0 & 0 & 0 & 0 & h_4 \\ 0 & h_4 & h_3 & h_2 & h_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_4 & h_3 & h_2 & h_1 & 0 \\ h_1 & 0 & 0 & 0 & 0 & h_4 & h_3 & h_2 \\ g_3 & g_2 & g_1 & 0 & 0 & 0 & 0 & g_4 \\ 0 & g_4 & g_3 & g_2 & g_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_4 & g_3 & g_2 & g_1 & 0 \\ g_1 & 0 & 0 & 0 & 0 & g_4 & g_3 & g_2 \end{pmatrix}$$

Matice \mathbf{W}^{-1} pro různou hloubku dekompozice J je tvaru:

$$\mathbf{W}_{1}^{-1} = \begin{bmatrix} \mathbf{H}_{1}^{\mathrm{T}} & \mathbf{G}_{1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\
\mathbf{W}_{2}^{-1} = \begin{bmatrix} \mathbf{H}_{1}^{\mathrm{T}} \mathbf{H}_{2}^{\mathrm{T}} & \mathbf{H}_{1}^{\mathrm{T}} \mathbf{H}_{2}^{\mathrm{T}} & \mathbf{H}_{1}^{\mathrm{T}} \mathbf{G}_{2}^{\mathrm{T}} & \mathbf{G}_{1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\
\mathbf{W}_{3}^{-1} = \begin{bmatrix} \mathbf{H}_{1}^{\mathrm{T}} \mathbf{H}_{2}^{\mathrm{T}} \mathbf{H}_{3}^{\mathrm{T}} & \mathbf{H}_{1}^{\mathrm{T}} \mathbf{H}_{2}^{\mathrm{T}} \mathbf{G}_{3}^{\mathrm{T}} & \mathbf{H}_{1}^{\mathrm{T}} \mathbf{G}_{2}^{\mathrm{T}} & \mathbf{G}_{1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\
\vdots \\
\mathbf{W}_{J}^{-1} = \begin{bmatrix} \mathbf{H}_{1}^{\mathrm{T}} \cdots \mathbf{H}_{J}^{T} & \mathbf{H}_{1}^{\mathrm{T}} \cdots \mathbf{H}_{J-1}^{\mathrm{T}} \mathbf{G}_{J}^{T} & \dots & \mathbf{H}_{1}^{\mathrm{T}} \mathbf{G}_{2}^{\mathrm{T}} & \mathbf{G}_{1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$
(2.10)

Z hlediska vysoké výpočtové náročnosti není násobení vektoru \mathbf{x} s maticí příliš výhodné (kvadratická složitost), lze jej nahradit Mallatovým algoritmem. Okrajové jevy se zde řeší dodefinováním hodnot na okraji. Princip algoritmu spočívá v tom, že vstupní vektor \mathbf{x} filtrujeme přes \mathbf{h} a \mathbf{g} a z výsledných posloupností vypustíme každý druhý vzorek (*dyadické podvzorkování* ozn. \downarrow 2). Takto získáme dva druhy koeficientů:

- aproximační waveletové koeficienty a koeficienty získané pomocí filtru typu dolní propust
- detailní waveletové koeficienty d koeficienty získané pomocí filtru typu horní propust

Detailní koeficinty uchováme. V další iteraci postup opakujeme, vstup tvoří vektor získaných aproximačních vlnkových koeficientů. Počet iterací udává hloubku dekompozice. Maximální hloubku dekompozice budeme uvažovat v závislosti na délce



Obr. 2.1: Ukázka Mallatova algoritmu pro hloubku dekompozice J = 3.

signálu s jako $J_{max} = \log_2 s$.

Pro 2D signál můžeme matici vytvořit podobným způsobem. V první iteraci provedeme filtraci oběma filtry nejprve po řádcích a následně po sloupcích. Obdržíme čtyři druhy koeficientů: aproximační koeficenty a detailní koeficienty v horizontálním, vertikálním a diagonálním směru. Vstupem do další iterace je opět vektor aproximačních koeficientů.

2.3.2 Trénované slovníky

Otázka nalezení optimálního slovníku **A** začala být studována teprve nedávno roku 1996 [7]. Vychází se z předpokladu, že pro známou kategorii signálu $\{\mathbf{y}_i\}_{i=1}^M$, lze nalézt takový slovník, ve kterém budou tyto signály co nejřidší. V tomto případě předpokládáme, že odchylka ϵ od modelu, který nám databázi $\{\mathbf{y}_i\}_{i=1}^M$ generuje, je známá a úlohou je odhadnout slovník **A**.

Uvažujme následující optimalizační úlohu [1]:

$$\min_{\mathbf{A}, \{\mathbf{x}_i\}_{i=1}^M} \sum_{i=1}^M \|\mathbf{x}_i\|_0 \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i\|_2 \le \epsilon, \quad 1 \le i \le M,$$
(2.11)

kde \mathbf{y}_i popisuje daný signál jako nejřidší možnou reprezentaci vektoru \mathbf{x}_i nad neznámým slovníkem \mathbf{A} .

Daný problém lze řešet i jako úlohu:

$$\min_{\mathbf{A}, \{\mathbf{x}_i\}_{i=1}^M} \sum_{i=1}^M \|\mathbf{y}_i - \mathbf{A}\mathbf{x}_i\|_2^2 \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{x}_i\|_0 \le k_0, \quad 1 \le i \le M.$$
(2.12)

Řešením této úlohy je slovník **A**, který má pro všechny složky řídkost k_0 .

V příloze uvádíme dva základní algoritmy pro hledání optimálního slovníku **A**. Metoda optimálních směrů a metoda K-SVD.

2.4 Aproximativní metody

Uvažujme spark(\mathbf{A}) > 2k, kde existuje k-řídké řešení soustavy (P0), pak toto řešení je nejřidší možné a jednoznačné. Pro určení přesného řešení bychom ovšem museli porovnat $\binom{n}{k}$ kombinací podmnožin atomů matice \mathbf{A} . Tento postup odpovídá NP-složitosti a je v reálném čase pro větší n neřešitelný.

Pokud budeme chtít realizovat výpočet v přijatelném čase - alespoň polynominálním, přesné řešení budeme muset nahradit řešením přibližným. Za tímto účelem bylo vyvinuto množství aproximativních metod.

2.4.1 Relaxační algoritmy

Relaxační algoritmy vycházejí z myšlenky nahradit nekonvexní normu ℓ_0 konvexní ℓ_p normou. Tato náhrada umožní aplikovat na původní úlohu (P0) metody konvexní optimalizace. Za určitých podmínek se dostaneme k rešení přesnému nebo alespoň relativně blízkému.

ℓ_1 relaxace

Konvexní optimalizace nabízí celou škálu metod a algoritmů pro řešení minimalizačních úloh. Vyvstává proto přirozená otázka, zda je možné úlohu (P0) formulovat pomocí konvexní normy a tyto metody na daný problém využít. Použítím ℓ_1 normy přechází (P0) na úlohu:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem k} \quad \mathbf{A}\mathbf{x} = \mathbf{y}, \tag{P1}$$

Úloha (P1) dovoluje použít množství metod konvexní optimalizace. Za určitých podmínek se řešení úloh (P0) a (P1) opravdu shoduje.

V případě šumu se povoluje odchylka od přesného řešení, tedy

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem k} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_{2} \le \epsilon.$$
(P1 ϵ)

Podmínky ekvivalence řešení ℓ_0 a ℓ_1 - minimalizace

Již v předešlé kapitole bylo uvedeno, že v případě splnění podmínky (2.5) lze úlohu (P0) řešit ℓ_1 - minimalizací. Ekvivalence řešení úlohy v ℓ_0 a ℓ_1 normě tedy závisí

na vlastnostech matice **A**. Jednou z takovýchto vlastností je splnění konceptu *NSP* (Null Space Property) – vlastnosti nulového prostoru.

Označení 2.7. Označme $\mathbf{x}_T \in \mathbb{C}^n$, $T \subset \{1, \ldots n\}$ vektor odvozený z vektoru $\mathbf{x} \in \mathbb{C}^n$ tak, že prvky na pozicích patřící do množiny T zachováme a ostatní položíme nule. $T^c = \{1, \ldots n\} \setminus T$ značí komplement množiny T.

Definice 2.8. [1] Matice $\mathbf{A} \in \mathbb{C}^{m \times n}$ splňuje *NSP* řádu k s konstantou $\gamma \in (0, 1)$, pokud platí

$$\|\eta_T\|_1 \le \gamma \|\eta_{T^C}\|_1 \tag{2.13}$$

pro všechny množiny $T \subset \{1, \ldots, n\}$, kde |T| < k, a pro všechny vektory $\eta \in \ker(\mathbf{A})$.

Pokud matice **A** tuto podmínku splňuje, je možno k-řídké řešení nalézt ℓ_1 - minimalizací. Toto řešení je navíc i jednoznačné. Chyba řešení je přitom omezena shora. Jelikož reálné signály většinou nejsou přímo řídké, ale obsahují malé nenulové hodnoty, vystupuje je zde $\sigma_k(\mathbf{x})_1$.

Tvrzení 2.9. [5] Nechť matice $\mathbf{A} \in \mathbb{C}^{m \times n}$ splňuje NSP řádu k s konstantou $\gamma \in (0,1)$. Nechť $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{y} = \mathbf{A}\mathbf{x}$ a $\mathbf{x}^* \in \mathbb{C}^n$ je řešením ℓ_1 minimalizace. Potom

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \le \frac{2(1+\gamma)}{1-\gamma} \sigma_k(\mathbf{x})_1.$$
(2.14)

 $d\hat{u}kaz:[5]$

Další možností, jak ověřit, zda matice A má potřebný tvar je koncept *RIP* (Restricted Isometry Property) – vlastnost zeslabené izometrie. Oproti *NSP* poskytuje výpočetně přijatelnější řešení a je navíc i stabilní pod vlivem šumu.

Definice 2.10. [5] Konstanta zeslabené izometrie matice $\mathbf{A} \in \mathbb{C}^{m \times n}$ je nejmenší číslo δ_k takové, že platí

$$(1 - \delta_k) \le \frac{\|\mathbf{A}\mathbf{z}\|_2^2}{\|\mathbf{z}\|_2^2} \le (1 + \delta_k)$$
 (2.15)

pro všechny vektory $\mathbf{z} \in \Sigma_k^n$. Řekneme, že matice **A** splňuje *RIP* řádu k s konstantou δ_k , pokud $\delta_k \in (0, 1)$.

Princip zeslabené izometrie spočívá v omezení se pouze na všechny podmatice \mathbf{A} o k sloupcích, kde nevyžadujeme přesnou izometrii, ale povolujeme odchylku δ_k . Tedy všechny takové matice budou přibližně ortogonální. Obecně uvažujeme všechny podmatice o maximálně k sloupcích, neboť dopředu nejsme schopni odhadnout, které prvky vektoru \mathbf{x} budou nenulové a tedy i které atomy matice \mathbf{A} se budou podílet na reprezentaci signálu y.

Konstantu zeslabené izometrie lze určit přímým výpočtem[5]:

$$\delta_k = \max_{T \subset \{1,\dots,n\}, |T| < k} \left\| \mathbf{A}_T^T \mathbf{A}_T - \mathbf{I} \right\|_{2 \to 2}$$
(2.16)

Lze dokázat, že vztah meziRIP a NSP je jednoznačně určen:

Tvrzení 2.11. [5] Nechť $\mathbf{A} \in \mathbb{C}^{m \times n}$ splňuje RIP řádu K = k + h s konstantou $\delta_k \in (0, 1)$. Pak \mathbf{A} spňuje NSP řádu k s konstantou

$$\gamma = \sqrt{\frac{k}{h} \frac{1 + \delta_K}{1 - \delta_K}} \tag{2.17}$$

 $d\hat{u}kaz:[5]$

Na základě vlastností *RIP* můžeme omezit chybu aproximace.

Tvrzení 2.12. [5] Nechť $\mathbf{A} \in \mathbb{C}^{m \times n}$ splňuje RIP řádu 3k s konstantou $\delta_3 k < \frac{1}{3}$. Pro $\mathbf{x} \in \mathbb{C}^n$, nechť $\mathbf{y} = \mathbf{A}\mathbf{x}$ a $x^* \in \mathbb{C}^n$ je řesení ℓ_1 -minimalizace. Pak

$$\|\mathbf{x} - \mathbf{x}^*\|_2 \le C \frac{\sigma_k(\mathbf{x})_1}{\sqrt{k}} \tag{2.18}$$

kde C je konstanta závisející pouze na $\delta_3 k$.

d u kaz: [5]

Tvrzení 2.13. [5] Předpokládejme, že matice $\mathbf{A} \in \mathbb{C}^{m \times n}$ vyhovuje RIP řádu 2k s konstantou

$$\delta_2 k \le \frac{2}{3 + \sqrt{\frac{7}{4}}} \approx 0,4627.$$
(2.19)

Následující vztahy platí pro všechna $\mathbf{x} \in \mathbb{C}^n$. Předpokládejme, že naměřená data jsou zatížena šumem, tedy $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, $\|\mathbf{e}\|_2 \leq \epsilon$, \mathbf{x}^* je řešení úlohy

$$\min_{\mathbf{z}} \|\mathbf{z}\|_{1} \quad vzhledem \ k \quad \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_{2} \le \epsilon.$$
(2.20)

Potom

$$\left\|\mathbf{x} - \mathbf{x}^*\right\|_2 \le C_1 \frac{\sigma_k \left(\mathbf{x}\right)_1}{\sqrt{k}} + C_{2^{\epsilon}},\tag{2.21}$$

kde kladné konstanty C_1 , C_2 závisejí pouze na δ_{2k} .

důkaz:[5]

Jak lze vidět, celkovou chybu můžeme rozdělit na část závisející pouze na řídkosti k a na část podmíněnou energií rušení – šum.

3 POTLAČOVÁNÍ ŠUMU V ŘÍDKÉM SIGNÁLU

Jednou z aplikací aparátu řídké reprezentace je potlačení šumu v řídkém signálu. Jedná se o řešení již zmiňované úlohy (P0 ϵ). Provedeme-li ℓ_1 -relaxaci, dostaneme úlohu (P1 ϵ) známou též jako BPDN (*Basis Pursuit Denoising*). Úlohu lze formulovat i v jiném tvaru ℓ_1 -relaxace: LASSO (*Least Absolute Selection and Shrinkage Operator*) nebo QP problém (*Quadratic Programming problem*)[11].

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_{2} \le \epsilon, \tag{BPDN}$$

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{x}\|_1 \le \epsilon, \tag{LASSO}$$

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_{2}^{2} \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{x}\|_{1} \le \epsilon.$$
 (QP)

Úlohy lze přepsat i bez omezující podmínky za pomoci regulačního parametru $\lambda > 0$, který dává do souvislosti řídkost řešení a přesnost vzhledem ke kvadratické chybě $\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$.

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{2}^{2} + \lambda \|\mathbf{x}\|_{1}$$
 (QP λ)

Mezi metody pro hledání přesného řešení dané úlohy patří např. iterační algoritmus In-Crowd [10], FISTA [6] aj.

3.1 Zobecnění úlohy

Úlohu (QP λ) lze formulovat jako obecný konvexní optimalizační problém tvaru

$$\min_{x \in \mathbb{R}^N} f_1(x) + \ldots + f_m(x), \tag{ZU}$$

kde $f_1(x) + \ldots + f_m(x) \in \mathbb{R}^N$ jsou konvexní funkce. Hlavním problém při řešení této úlohy pramení z požadavku na diferencovatelnost funkcí $f_1(x) + \ldots + f_m(x)$. Takto silný požadavek je v praktických úlohách nereálné splnit. Abychom mohli použít některou z hladkých optimalizačních technik, zavedeme si pojem *proximity operátor*, který nám umožní pracovat i s nediferencovatelnými funkcemi.

Definice 3.1. Nechť množina $C \subset \mathbb{R}^N$, pak *charakteristická funkce konvexní množiny* C je dána

$$\iota_C : \mathbb{R}^N \to \{0, \infty\} : x \mapsto \begin{cases} 0 \text{ pokud } x \in C \\ \infty \text{ jinak} \end{cases}$$

Definice 3.2. Nechť množina $C \subset \mathbb{R}^N$. Pak vzdálenost funkce od neprázdné množiny C je dána

$$d_C: \mathbb{R}^N \to \langle 0, \infty \rangle: \mathbf{x} \mapsto \inf_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_2.$$

Definice 3.3. Nechť $\mathbf{x} \in \mathbb{R}^N$ a $C \subset \mathbb{R}^N$. Pak *projekcí* \mathbf{x} na neprázdnou uzavřenou konvexní množinu C rozumíme bod $P_C x$ takový, že

$$d_C(\mathbf{x}) = \|\mathbf{x} - P_C \mathbf{x}\|_2$$

Definice 3.4. Necht $\Gamma_0(\mathbb{R}^N)$ je třída zdola polospojitých konvexních funkcí z \mathbb{R}^N do $(-\infty, \infty)$ takových, že dom $f \neq \emptyset$.

Jeden z prvních široce užívaných algoritmů konvexní optimalizace v oblasti zpracování signálu byl POSC (*Projection Onto Convex Sets*), tedy projekce na konvexní množinu. Tento algoritmus byl podmíněn několika konvexními omezeními, pod kterými se původní úloha zjednodušila na hledání

$$\mathbf{x} \in \bigcap_{i=1}^m C_i,$$

kde C_i jsou konvexní množiny zapojované jednotlivě přes projekční operátory P_{C_i} . Tedy platí

$$x_{n+1} = P_{C_1} \dots P_{C_m} x_n.$$

Takto daná projekce P_{C_i} je řešením problému:

$$\min_{\mathbf{y}\in\mathbb{R}^N}\iota_{C_i}(\mathbf{y})+\tfrac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2.$$

Tento problém byl v roce 1962 zobecněn a projekční operátor byl nahrazen libovolnou funkcí $f_i \in \Gamma_0(\mathbb{R}^N)$.

Definice 3.5. Nechť $f_i \in \Gamma_0(\mathbb{R}^N)$. Pro všechna $x \in \mathbb{R}^N$ optimalizační problém

$$\min_{\mathbf{y} \in \mathbb{R}^N} f_i(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$$
 (Prox)

připouští jediné řešení, které označíme $\operatorname{prox}_{f_i} \mathbf{x}$. Takto definovaný operátor

$$\operatorname{prox}_{f_i} : \mathbb{R}^N \to \mathbb{R}^N,$$

nazýváme proximity operátor funkce f_i .

Proximity operátor má spoustu vlastností výhodných pro řešení optimalizačnách úloh, např.

$$\left(\forall \mathbf{x} \in \mathbb{R}^{N}\right) \left(\forall \mathbf{y} \in \mathbb{R}^{N}\right) \quad \left\| \operatorname{prox}_{f_{i}} x - \operatorname{prox}_{f_{i}} y \right\|_{2} \leq \|\mathbf{x} - \mathbf{y}\|_{2},$$

a mnoho dalších [8].

Přirozený výklad proximity operátoru popisuje přímo problematika odšumování. Předpokládejme klasickou úlohu odšumování. Pro hledaný signál $\overline{\mathbf{x}} \in \mathbb{R}^N$ obdržíme úlohu

$$\mathbf{y} = \overline{\mathbf{x}} + \mathbf{w},$$

kde **w** modeluje šum. Tuto úlohu můžeme chápat jako (Prox), kde $\frac{1}{2} \|\cdot - y\|_2^2$ představuje věrohodnost dat a f_i zahrnuje naše znalosti, které máme o hledaném signálu $\overline{\mathbf{x}}$.

3.1.1 Algoritmus Forward-Backward

Uvažujme následující úlohu[8]:

$$\begin{split} \min_{x \in \mathbb{R}} f_1(x) + f_2(x), \quad (3.1) \\ \text{kde } f_1 \in \Gamma_0(\mathbb{R}^N), \ f_2 : \mathbb{R}^N \to \mathbb{R} \text{ je konvexní funkce a } \nabla f_2 \text{ je} \\ \beta - \text{Lipschitzovsky spojitý tzn. pro } \forall (x, y) \in \mathbb{R}^N \times \mathbb{R}^N, \ \beta > 0 \\ \|\nabla f_2(x) - \nabla f_2(y)\|_2 \leq \beta \|x - y\|_2 \\ \text{Předpokládejme } f_1(x) + f_2(x) \to \infty, \text{ pokud } \|\mathbf{x}\|_2 \to \infty. \end{split}$$

Daná úloha připouští právě jedno řešení, které je dáno pro každé $\gamma>0$ jako pevný bod

$$x = \operatorname{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)).$$

Iteračně obdržíme tvar[8]:

$$x_{n+1} = \underbrace{\operatorname{prox}_{\gamma_n f_1}}_{\operatorname{zpětný}} \underbrace{(x_n - \gamma_n \nabla f_2(x_n))}_{\operatorname{dopředný krok}}.$$
(3.2)

kde γ_n představuje velikost kroku z ohraničeného intervalu.

V případě, kdy $f_1 = 0$, přechází úloha v gradientní metodu pro hledání minima Lipschitzovsky diferencovatelné funkce

$$x_{n+1} = x_n - \gamma_n \nabla f_2(x_n)$$

Pokud $f_2 = 0$, úloha se redukuje na *proximal point algoritmus* pro hledání minima nediferencovatelné funkce

 $x_{n+1} = \operatorname{prox}_{\gamma_n f_1} x_n$

Forward-Backward algoritmus lze považovat za kombinaci těchto dvou základních schémat. Po zahrnutí relaxačního parametru $(\lambda_n)_{n\leq 0}$ nabývá algoritmus tvaru

Forward-Backward Algoritmus

Pro pevně zvolené $\epsilon \in (0, \min\left\{1, \frac{1}{\beta}\right\}), x_0 \in \mathbb{R}^N$ pro n = 0, 1, 2... $\gamma_n \in \left\langle\epsilon, \frac{2}{\beta} - \epsilon\right\rangle$ $y_n = \gamma_n \nabla f_2(x_n)$ $\lambda_n \in \langle\epsilon, 1\rangle$ $x_{n+1} = x_n + \lambda_n(\operatorname{prox}_{\gamma_n f_1} y_n - x_n)$

Tvrzení 3.6. [12] Každá posloupnost $(x_n)_{n\leq 0}$ generována (3.3) konverguje k řešení problému (3.1).

(3.3)

důkaz:[12]

Forward-Backward Algoritmus pro konstantní krok Pro pevně zvolené $\epsilon \in (0, \frac{3}{4}), x_0 \in \mathbb{R}^N$ pro n = 0, 1, 2... $y_n = x_n - \beta^{-1} \nabla f_2(x_n)$ $\lambda_n \in \left\langle \epsilon, \frac{3}{2} - \epsilon \right\rangle$ (3.4) $x_{n+1} = x_n + \lambda_n (\operatorname{prox}_{\beta^{-1}f_1} y_n - x_n)$

Tvrzení 3.7. [12] Každá posloupnost $(x_n)_{n\leq 0}$ generována (3.4) konverguje k řešení problému (3.1).

důkaz:[12]

Ačkoliv je konvergence k řešení dokázána, nemáme zaručeno, že rychlost konvergence bude dostačující, aby byla úloha numericky řešitelná.

3.1.2 Algoritmus Douglas-Rachford

Uvažujme následující úlohu:

$$\begin{split} \min_{x \in \mathbb{R}} f_1(x) + f_2(x), \quad (3.5) \end{split}$$
kde funkce $f_1, f_2 \in \Gamma_0(\mathbb{R}^N)$ takové, že $(ri \operatorname{dom} f_1) \cap (ri \operatorname{dom} f_2) \neq \emptyset$ Předpokládejme $f_1(x) + f_2(x) \to \infty$, pokud $\|x\|_2 \to \infty$.

Daná úloha připouští právě jedno řešení, které je charakterizováno pro každé $\gamma > 0$ dvoustupňovou podmínkou [9]:

$$\begin{cases} x = \operatorname{prox}_{\gamma f_2} y \\ \operatorname{prox}_{\gamma f_2} y = \operatorname{prox}_{\gamma f_1}(2\operatorname{prox}_{\gamma f_2} y - y) \end{cases}$$

Po zahrnutí relaxačního parametru $(\lambda_n)_{n \leq 0}$ nabývá algoritmus tvaru

Douglas-Rachford Algoritmus Pro pevně zvolené $\epsilon \in (0, 1), y_0 \in \mathbb{R}^N$ pro n = 0, 1, 2... $x_n = \operatorname{prox}_{\gamma f_2} y_n$ $\lambda_n \in \langle \epsilon, 2 - \epsilon \rangle$ $y_{n+1} = y_n + \lambda_n (\operatorname{prox}_{\gamma f_1} (2x_n - y_n) - x_n)$ (3.6)

Poznámka 3.8. V případě $\lambda_n = 2$ mluvíme o Peaceman-Rachford algoritmu. Pro zaručení konvergence jsou nutny další předpoklady [13].

Tvrzení 3.9. [9] Každá posloupnost $(x_n)_{n\leq 0}$ generována (3.6) konverguje k řešení problému (7.4).

důkaz:[9]

Narozdíl od Forward-Backward algoritmu nevyžaduje Douglas-Rachford algoritmus splnění podmínky Lipschitzovské diferencovatelnosti. Může být ale numericky náročnější, neboť v každé iteraci se provádí dva proximální kroky, zatímco v případě Forward-Backward algoritmu pouze jeden. Záleží ale na implementaci, nelze obecně rozhodnout, který z nich je účinější.

4 DANTZIG SELEKTOR

Dantzig Selektor je další možnou formulací úlohy pro hledání řídkého řešení. Tento algoritmus byl navžen teprve nedávno (2007) v práci [14]. Název Dantzig má na počest tvůrce simplexové metody George Dantziga a jak si ukážeme, k lineárnímu programování má velmi blízko.

Narozdíl od formulace (BPDN), kde hledáme nejřidší možné řešení jako

$$\mathbf{x}_{BPDN} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem k } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{2} \le \epsilon,$$
(4.1)

v případě Dantzig Selektoru dostáváme

$$\mathbf{x}_{DS} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem k } \|\mathbf{A}^{\mathrm{T}}(\mathbf{y} - \mathbf{A}\mathbf{x})\|_{\infty} \leq \lambda \sigma.$$
 (DS)

Uvažujme případ, že byl výstup zanesen náhodným Gaussovským bílým šumem \mathbf{e} s rozdělením $e_i \sim \mathcal{N}(0, \sigma^2)$, tedy $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{e}$, naším cílem je spolehlivě odhadnout \mathbf{x}_0 . Podívejme se, jak jednotlivé algoritmy (4.1) a (DS) odhadují případná rezidua.

Jako přirozené nám připadne omezení podmínkou $\|\mathbf{y} - \mathbf{A}\mathbf{x}\| \leq \epsilon$, neboť reziduum očekáváme v nějaké omezené formě. Podíváme-li se však na toto omezení podrobněji, nic nám neprozradí, jaké podoby může rezidum $\mathbf{r} = \mathbf{y} - \mathbf{A}\mathbf{x}$ nabývat a nevynutí nám tvar v podobě náhodného bílého šumu. Dokonce připouští i silně korelovaná rezidua, ačkoliv jsou poměrně nepravděpodobná.

Naopak v případě (DS), pokud se podíváme na tvar vztažné podmínky, jedná se o vnitřní součin mezi složkami reziduí a sloupci matice **A**, který má být pod předem specifikovanou hodnotu $\lambda\sigma$. Právě toto nám zaručí, že žádná hodnota z rezidua nebude dominantní a tady nevzniknou výrazné korelace, které by nám povahu rezidua ovlivnili. Všechny složky **r** budou dle očekávání malé.

Problém (DS) lze přidáním pomocné proměnné $\mathbf{u} \ge |\mathbf{x}|$ přeformulovat do podoby lineárního programovaní:

$$\min_{\mathbf{u},\mathbf{x}} \mathbf{1}^{\mathrm{T}} \mathbf{u} \quad \text{vzhledem k} \begin{cases} -\mathbf{u} \leq \mathbf{x} \leq \mathbf{u} \\ \text{a zároveň} \\ -\sigma\lambda \mathbf{1} \leq \mathbf{A}^{\mathrm{T}} \left(\mathbf{b} - \mathbf{A} \mathbf{x} \right) \leq \sigma\lambda \mathbf{1}, \end{cases}$$

kde 1 je vektor jedniček.

Všechna omezení jsou lineární. Můžeme tedy na tento problém využít množství řešičů z oblasti lineárního programování (např. Iterační Shrinkage algoritmus [1]).

Jiný způsob, jak přepsat (DS) na úlohu lineárního programování je rozdělit \mathbf{x} na

dva vektory v podobě $\mathbf{x} = \mathbf{u} - \mathbf{v}$, kde pro vektory \mathbf{u} , \mathbf{v} platí $\mathbf{u}^{\mathrm{T}}\mathbf{v} = 0$. Úloha pak může být formulována jako :

$$\min_{\mathbf{u},\mathbf{v}} \mathbf{1}^{\mathrm{T}}\mathbf{u} + \mathbf{1}^{\mathrm{T}}\mathbf{v} \quad \text{vzhledem } \mathbf{k} \begin{cases} \mathbf{u} \leq 0, \quad \mathbf{v} \leq 0 \\ \text{a zároveň} \\ -\sigma\lambda \mathbf{1} \leq \mathbf{A}^{\mathrm{T}} \left(\mathbf{b} - \mathbf{A}\mathbf{u} + \mathbf{A}\mathbf{v} \right) \leq \sigma\lambda \mathbf{1} \end{cases}$$

4.1 Ekvivalence řešení BPDN a DS

Na první pohled by se zdálo, že formulace DS má předpoklady pro lepší a kvalitnější rekonstrukci signálu než v případě BPDN. Ačkoliv jsou její podmínky daleko více informativní, nemusí to být jasným pravidlem. V případě, kdy je matice **A** maticí unitární, jsou úlohy (4.1) a (DS) dokonce ekvivalentní [1].

Využijeme vlastnosti, že v případě unitární matice platí $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}$. Omezení v (DS) můžeme díky tomuto upravit roznásobením na $\|\mathbf{A}^{T}\mathbf{y} - \mathbf{x}\|_{\infty} \leq \lambda \sigma$. Díky tomuto můžeme DS formulovat jako množinu *m* nezávislých optimalizačních úloh tvaru:

$$\min_{\mathbf{x}} |x| \quad \text{vzhledem k} \left| \mathbf{a}_i^{\mathrm{T}} \mathbf{y} - x \right| \le \lambda \sigma, \quad i = 1, 2, \cdots, m.$$
(4.2)

Řešení pak obdržíme ve formě:

$$x_{i} = \begin{cases} \mathbf{a}_{i}^{\mathrm{T}} \mathbf{y} - \lambda \sigma & \mathbf{a}_{i}^{\mathrm{T}} \mathbf{y} \geq \lambda \sigma \\ 0 & \left| \mathbf{a}_{i}^{\mathrm{T}} \mathbf{y} \right| < \lambda \sigma \\ \mathbf{a}_{i}^{\mathrm{T}} \mathbf{y} + \lambda \sigma & \mathbf{a}_{i}^{\mathrm{T}} \mathbf{y} \leq -\lambda \sigma \end{cases}$$

Toto řešení odpovídá stejně jako řešení pro (4.1) jemnému prahování - pro vhodnou volbu $\lambda \sigma$ k ϵ . V obecném případě tato ekvivalence ovšem neplatí.

4.2 Podmínky řešitelnosti

V případě DS budeme na matici **A** klást silnější požadavek než je RIP. Požadujeme, aby matice **A** vyhovovala konceptu *ROP* (Restricted Orthogonality Property)– vlastnost zeslabené ortogonality:

Definice 4.1. [1] Nechť matice $\mathbf{A} \in \mathbb{C}^{n \times m}$ (m > n) má normované sloupce (v ℓ_2 normě). Pro $k_1, k_2 < n$, kde $k \in \mathbb{N}$, uvažujme dvě disjunktní množiny vytvořené z k_1 a k_2 sloupců matice \mathbf{A} a označme je jako podmatice \mathbf{A}_{k_1} a \mathbf{A}_{k_2} . Konstantu zeslabené ortogonality definujeme jako nejmenší číslo θ_{k_1,k_2} takové, kde pro $\forall \mathbf{c}_1 \in \mathbb{R}^{k_1}$, $\forall \mathbf{c}_2 \in \mathbb{R}^{k_2}$ a pro všechny kombinace k_1, k_2 sloupců platí

$$\left|\mathbf{c}^{\mathrm{T}}\mathbf{A}_{k_{1}}^{\mathrm{T}}\mathbf{A}_{k_{2}}\mathbf{c}_{2}\right| = \left|\left\langle\mathbf{A}_{k_{1}}\mathbf{c}_{1},\mathbf{A}_{k_{2}}\mathbf{c}_{2}\right\rangle\right| \le \theta_{k_{1},k_{2}} \left\|\mathbf{c}_{1}\right\|_{2} \left\|\mathbf{c}_{2}\right\|_{2}.$$
(4.3)

Matice **A** spňující tato omezení má vlastnost k_1, k_2 -zeslabené ortogonatity s konstantou θ_{k_1,k_2} .

Stejně jako v případě RIP, přesný výpočet hodnot θ_{k_1,k_2} je nepraktický a zbytečně náročný. Vystačíme si s odhadem pomocí vzájemné koherence : $\theta_{k_1,k_2} \leq \sqrt{k_1k_2}\mu(\mathbf{A})$. Příklad matice vyhovující ROP jsou podobně jako u RIP náhodné matice.

Tvrzení 4.2 (Stabilita DS). Uvažujme problém (DS) definovaný jako trojici ($\mathbf{A}, \mathbf{y}, \sigma$) $a \ \lambda = \sqrt{2(1+a)\log m}$. Předpokládejme, že vektor $\mathbf{x}_0 \in \mathbb{R}^m$ je k-řídký a splňuje omezení $\delta_{2k} + \theta_{k,2k} < 1$, $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{e}$, kde \mathbf{e} je nenulový náhodný Gaussův šum s rozptylem σ^2 . Pak pro řešení $\mathbf{x}_D^{\lambda}S$ platí

$$\left\|\mathbf{x}_{DS}^{\lambda} - \mathbf{x}_{0}\right\|_{2}^{2} \le 2\log m \cdot k \cdot \sigma^{2}$$

$$(4.4)$$

s pravděpodobností nepřesahující $1 - \left(\sqrt{\pi \log m}m^a\right)^{-1} a C = \frac{4}{1 - \delta_{2k} + \theta_{k,2k}}.$

4.3 Variace a řešiče DS

Variací Dantzig selektoru, která obvykle přináší vyšší statistickou přesnost, je Gauss-Dantzig selektor [14]. Gauss-Dantzig selektor řeší případ, kdy by matice **A** byla ortogonální a formulace úlohy by přešla na problematiku měkkého prahování. V dvou krocích se provede úprava:

- odhadneme $I = \{i \mid x_i \neq 0\}$ s $\hat{I} = \{i \mid \hat{x}_i \neq 0\}$
- vytvoříme odhad

$$\hat{\mathbf{x}}_{\hat{I}} = \left(\mathbf{A}_{\hat{I}}^{\mathrm{T}} \mathbf{A}_{\hat{I}}\right)^{-1} \mathbf{A}_{\hat{I}}^{\mathrm{T}} \mathbf{y}, \qquad (4.5)$$

množině zbývajících souřadnic přiřadíme nulu.

Přestože lze na problematiku DS po přeformulování úlohy použít nástoje lineárního programování, jsou konstruovány algoritmy pro přímé řešení této úlohy. Mezi takovéto řešiče patří LADMM (Linearized Alternating Direction Method)[15] a DASSO (Dantzig Selector with Sequential Optimization)[16]. V případě DASSO lze ukázat za určitých podmínek podobnost s LASSO [16], [17], což není nic překvapivého vzhledem k ukázané ekvivalenci mezi DS a BPDN.

5 DEKVANTIZACE ŘÍDKÉHO SIGNÁLU

Pojem kvantování se nejčastěji objevuje v souvislosti s digitalizací signálu, tedy v problematice převodu spojitého signálu na diskrétní, aby mohl být dále počítačově zpracován. Probíhá ve dvou fázích: *vzorkování* a *kvantování*.

Vzorkování probíhá tím způsobem, že vodorovnou osu signálu rozdělíme na rovnoměrné úseky a v každém úseku odebereme jeden vzorek ze signálu. Tím dostáváme místo spojitého signálu množinu diskrétních bodů. Interval mezi jednotlivými úseky nazýváme vzorkovací kmitočet. Pokud předem neznáme nějaké vlastnosti zpracovávaného signálu, musí vzorkovací kmitočet splňovat vzorkovací teorém. Z hlediska zpracování signálu je vzorkování přirozenou záležitostí, neb spojitý signál má nekonečně malé detaily pro jejichž zpracování by bylo potřeba nekonečného času.

Další fází je *kvantování*. Přesné hodnoty vzorků jsou touto operací nahrazeny se zvolenou konečnou přesností hodnotami kvantovanými. Rozhodovací úroveň přiřadí vzorku jeho kvantizační hladinu.



Obr. 5.1: Princip vzorkování a kvantování spojitého signálu. Rozsah "kvantizéru" je vymezen mezní rozhodovací úrovní V, která je rozdělena na N kvantizačních hladin šířky Δ .

Rozlišujeme dva druhy kvantování:

- lineární stejná šířka všech kvantizačních hladin viz obr 5.1
- nelineární nelineární šířka kvantizačních hladin

Kvantování je nevratná operace a vždy způsobí pokles kvality vzorkovaného signálu. Jako proces *dekvantizace* označíme snahu získat původní hodnoty vzorků před kvantováním. Tato operace by postrádala smysl bez předpokladu znalosti vlastností vstupního signálu. Dále budeme předpokládat řídký signál.



Obr. 5.2: Výstup dekvantizace, tedy signál $\tilde{\mathbf{x}}$, je odhadem původního vstupního signálu \mathbf{x} . V případě analogového signálu se před vzorkováním provádí filtrace typu dolní propust, aby se zajistilo splnění vzorkovacího teorému po celou dobu zpracovávání. Omezí se tím zároveň rozsah kvantizéru.

5.1 Formulace úlohy dekvantizace

Jako **x** označme přesné hodnoty koeficintů signálu získané po vzorkovaní, hodnoty signálu po kvantování označme \mathbf{y}_q . Předpokládáme, že hledaný signál je řídký v nějaké bázi **A**. Z úlohy o kvantování víme, že maximální vzdálenost vzorkované hodnoty a kvantované nepřekročí hodnotu rovnu $\Delta/2$. Na základě těchto znalostí můžeme využít myšlenky Dantzig selektoru a úlohu o dekvantizaci formulovat jako optimalizační úlohu tvaru:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1} \quad \text{vzhledem k} \quad \|\mathbf{y}_{q} - \mathbf{A}\mathbf{x}\|_{\infty} \leq \Delta/2.$$
 (AQ)

Na rozdíl od Dantzig selektoru neklademe požadavek na omezení korelace. Postačí nám, že všechny hodnoty rozdílu našeho odhadu a kvantovaných hodnot budou pod $\Delta/2$. Nenastane tedy situace, kdybychom **y** odhadli hodnotou příslušící jiné kvantizační hladině.

I v tomto případě můžeme tento optimalizační problém (AQ) přeformulovat do úlohy lineárního programování. Rozdělením \mathbf{x} na dva vektory \mathbf{u} a \mathbf{v} , kde platí $\mathbf{x} = \mathbf{u} - \mathbf{v}$, dostaneme :

$$\min_{\mathbf{u},\mathbf{v}} \mathbf{1}^T \mathbf{u} + \mathbf{1}^T \mathbf{v} \quad \text{vzhledem k} \begin{cases} -\mathbf{u} \leq 0, \quad \mathbf{v} \leq 0\\ \text{a zároveň}\\ -\frac{\Delta}{2} \mathbf{1} \leq (\mathbf{y}_q - \mathbf{A}\mathbf{u} + \mathbf{A}\mathbf{v}) \leq \frac{\Delta}{2} \mathbf{1}, \end{cases}$$

kde 1 je vektor jedniček a pro vektory $\mathbf{u},\,\mathbf{v}$ platí $\mathbf{u}^T\mathbf{v}=0.$

6 MORFOLOGICKÁ ANALÝZA KOMPONENT

Morfologická analýza komponent (MCA) vychází z myšlenky, že obraz je ve své podstatě lineární kombinací dvou navzájem různých složek:

- textura pravidelnosti vyskytující se napříč celým obrazem
- trend(cartoon) po částech hladká složka

V této kapitole budeme tedy na pozorovaný signál **y** nahlížet jako na superpozici dvou různých subsignálu \mathbf{y}_1 , \mathbf{y}_2 a šumu \mathbf{v} , tedy $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2 + \mathbf{v}$. Každý subsignál je generován jiným slovníkem \mathbf{A}_1 a \mathbf{A}_2 . Úlohou MCA je rozdělení signálu na tyto dvě složky. Uvažujme optimalizační úlohu [1]:

$$\min_{\mathbf{x}_1,\mathbf{x}_2} \|\mathbf{x}_1\|_0 + \|\mathbf{x}_2\|_0 \quad \text{vzhledem } \mathbf{k} \quad \|\mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2\|_2^2 \le \delta \quad (\text{MCA})$$

Výsledné řešení $(\mathbf{x}_1^{\delta}, \mathbf{x}_2^{\delta})$ nám bude generovat obě složky obrazu, tedy $\hat{\mathbf{y}}_1 = \mathbf{A}_1 \mathbf{x}_1^{\delta}$, $\hat{\mathbf{y}}_2 = \mathbf{A}_2 \mathbf{x}_2^{\delta}$. Parametr δ zahrnuje působení šumu a nepřesnosti modelu řídké reprezentace. V tomto případě mluvíme o tzv. syntetickém modelu.

Přepsáním rovnice si lze povšimout nápadné podobnosti s úlohou o odšumování signálu:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{0} \quad \text{vzhledem k} \quad \|\mathbf{y} - \mathbf{A}_{a}\mathbf{x}_{a}\|_{2}^{2} \leq \delta,$$
(6.1)
kde $\mathbf{A}_{a} = [\mathbf{A}_{1}, \mathbf{A}_{2}]$ a odpovídající reprezentace je $\mathbf{x}_{a}^{T} = [\mathbf{x}_{1}^{T}, \mathbf{x}_{2}^{T}].$

Také úlohu o odšumovaní můžeme brát jako úlohu dekompozice, kde jedna složka je šum a druhá čistý obraz. Hlavním důvodem, proč signál dělit na dvě různé složky, je především schopnost díky nim signál přesněji reprezentovat. Uvažujme jako příklad problematiku komprese založené na waveletech. Tento způsob velice dobře popíše obraz z hlediska její trendové složky, ale pro texturu již příliš vhodný není a tak může být celková komprese méně kvalitní. Pokud bychom ovšem provedli kompresi pro každou složku zvlášť, kde bychom využili specifik každé z nich a vhodně ji popsali, docílíme lepšího výsledku. Rovněž lze tento postup uplatnit v případě detekce hran v obraze nebo v případě dokreslování chybějících částí v obrazu [19].

MCA můžeme použít na obraz jako na celek, pak se jedná o globálním MCA. Nebo ve formě překrývajích se bloků, v tomto případě hovoříme o lokálním MCA. Obě úlohy si přiblížíme, ačkoliv využívat budeme pouze globální MCA.

MCA není ovšem jediný způsob jak rozdělit obraz na texturu a trend, jiný přístup lze nalézt např. v [18].

6.1 Globální MCA

Formulace úlohy

Uvažujme ideální obraz $\mathbf{y}_0 \in \mathbb{R}^N$ (tedy velikost $\sqrt{N} \times \sqrt{N}$ pixelů), který je složen ze dvou částí: trendu \mathbf{y}_c a textury \mathbf{y}_t , tj. $\mathbf{y}_0 = \mathbf{y}_c + \mathbf{y}_t$. Obraz je měřen za přítomnosti aditivního Gaussovského šumu \mathbf{v} se známou směrodatnou odchylkou. Naměřený obraz je pak

$$\mathbf{y} = \mathbf{y}_0 + \mathbf{v} = \mathbf{y}_c + \mathbf{y}_t + \mathbf{v}. \tag{6.2}$$

Cílem je navrhnout takový algoritmus, kde bychom získali zpět obě části ideálního obrazu tj. \mathbf{y}_c a \mathbf{y}_t . Použijeme formulaci (MCA) a s použitím ℓ_1 normy řešení navrhneme ve tvaru [1]:

$$\hat{\mathbf{x}}_{c}, \hat{\mathbf{x}}_{t} = \arg\min_{\mathbf{x}_{c}, \mathbf{x}_{t}} \lambda \|\mathbf{x}_{c}\|_{1} + \lambda \|\mathbf{x}_{t}\|_{1} + \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{c}\mathbf{x}_{c} - \mathbf{A}_{t}\mathbf{x}_{t}\|_{2}^{2}$$
(6.3)

Odhad hledaných hodnot \mathbf{y}_c a \mathbf{y}_t obdržíme jako $\hat{\mathbf{y}}_c = \mathbf{A}_c \hat{\mathbf{x}}_c$ a $\hat{\mathbf{y}}_t = \mathbf{A}_t \hat{\mathbf{x}}_t$, kde $\mathbf{A}_c \in \mathbb{R}^{N \times M_c}$, $\mathbf{A}_t \in \mathbb{R}^{N \times M_t}$ jsou matice voleny tak, aby umožnily řídkou reprezentaci dané složky obrazu.

- A_c volba závisí na typu kresby, kterou v obraze očekáváme; obvykle se jedná o slovníky waveletového typu, curvelets nebo coutourlets
- \mathbf{A}_t měla by obsahovat oscilační atomy, aby pokryla pravidelné útvary textury; nejčastěji využíváme DCT slovníky

Pokud bychom uvažovali speciální případ, kdy by matice \mathbf{A}_c , \mathbf{A}_t byly čtvercové a invertibilní, platilo by: $\hat{\mathbf{x}}_c = \mathbf{A}_c^{-1}\hat{\mathbf{y}}_c$, $\hat{\mathbf{x}}_t = \mathbf{A}_t^{-1}\hat{\mathbf{y}}_t$. Úlohu bychom mohli formulovat analyticky, tedy přímo pro odhady $\hat{\mathbf{y}}_c$ a $\hat{\mathbf{y}}_t$:

$$\hat{\mathbf{y}}_{c}, \hat{\mathbf{y}}_{t} = \arg\min_{\hat{\mathbf{x}}_{c}, \hat{\mathbf{x}}_{t}} \lambda \left\| \mathbf{A}_{c}^{-1} \mathbf{y}_{c} \right\|_{1} + \lambda \left\| \mathbf{A}_{t}^{-1} \mathbf{y}_{t} \right\|_{1} + \frac{1}{2} \left\| \mathbf{y} - \mathbf{y}_{c} - \mathbf{y}_{t} \right\|_{2}^{2}.$$
(6.4)

Za těchto podmínek by byly obě formulace ekvivalentní. V obecném případě by samozřejmě \mathbf{A}^{-1} bylo nahrazeno \mathbf{A}^+ . Podíváme-li se na obě úlohy, je zřejmé, že úloha (6.4) bude menší složitosti nežli úloha (6.3), neboť se hledá přímo $\hat{\mathbf{y}}_c$ a $\hat{\mathbf{y}}_t$ bez dalšího násobení matic. Úlohu je možné řešit kupříkladu iterativními shrinkage algoritmy [1].

6.2 Lokální MCA

Namísto práce s celým obrazem jako celkem, uvažujme jen jeho malé bloky (patches) **p** o velikosti $\sqrt{n} \times \sqrt{n}$ pixelů, kde $n \ll N$. Uspořádejme tyto bloky lexikograficky jako sloupcové vektory $\mathbf{p} \in \mathbb{R}^n$. Předpokládejme, že tyto bloky odpovídají nějakému řídkému modelu s tolerancí ϵ , tedy existuje takový slovník $\mathbf{A} \in \mathbb{R}^{n \times m}$ $(m \ge n)$, že každý blok **p** má svou odpovídající reprezentaci $\mathbf{q} \in \mathbb{R}^m$, splňující podmínku $\|\mathbf{A}\mathbf{q} - \mathbf{p}\|_2 \le \epsilon, \, \|\mathbf{q}\|_0 = k_0.$

Uvažujme ideální obraz $\mathbf{y}_0 = \mathbf{y}_c + \mathbf{y}_t$. Definujeme operátor $\mathbf{R}_k \in \mathbb{R}^{n \times M}$ jako operátor přístupu ke k-tému bloku - $\mathbf{p}_k^0 = \mathbf{R}_k \mathbf{y}_0$. Pro každý blok $\mathbf{R}_k \mathbf{y}_0 = \mathbf{R}_k \mathbf{y}_c + \mathbf{R}_k \mathbf{y}_t$ existují dva řídké vektory \mathbf{q}_c^k , \mathbf{q}_t^k takové, že

$$\left\|\mathbf{R}_{k}\mathbf{y}_{c}+\mathbf{R}_{k}\mathbf{y}_{t}-\mathbf{A}_{c}\mathbf{q}_{c}^{k}-\mathbf{A}_{t}\mathbf{q}_{t}^{k}\right\|_{2}\leq\epsilon,$$
(6.5)

kde \mathbf{A}_c a \mathbf{A}_t jsou pevné slovníky.

Výhodou lokální MCA je možnost paralelní implementace a možnosti využití trénovaných slovníků. I v tomto případě lze nalézt podobnost s algoritmy na odšumování, a proto výpočet \mathbf{y}_c a \mathbf{y}_t odvodíme z formulace K-SVD algoritmu pro odšumování (jedná se o lokální odšumování a využitím trénovaného slovníku K-SVD) [1].

Výraz (6.5) přepíšeme na $\|\mathbf{R}_k \mathbf{y}_0 - \mathbf{A}_a \mathbf{q}_a^k\|_2 \leq \epsilon$ a minimalizujeme odhad chyby pomocí odhadu MAP (Maximum a posteriori probability):

$$\left\{\left\{\hat{\mathbf{q}}_{a}^{k}\right\}_{k=1}^{N}, \hat{\mathbf{y}}\right\} = \arg\min_{\mathbf{z}, \left\{\mathbf{q}_{a}^{k}\right\}_{k}} \lambda \left\|\mathbf{z} - \mathbf{y}\right\|_{2}^{2} + \sum_{k=1}^{N} \left\|\mathbf{q}_{a}^{k}\right\|_{0} + \sum_{k=1}^{N} \left\|\mathbf{A}_{a}\mathbf{q}_{a}^{k} - \mathbf{R}_{k}\mathbf{z}\right\|_{2}^{2}, \quad (6.6)$$

kde **y** je měřený obraz, **z** je odšuměný obraz. První člen zajištuje, že obrazy **y** a **z** si budou blízké. Další členy zaručují správnou rekonstrukci obrazu **z**. Každý blok $\mathbf{p}_k = \mathbf{R}_k \mathbf{z}$ o velikosti $\sqrt{n} \times \sqrt{n}$ má v každé pozici řídkou reprezentaci s omezenou chybou. Jednotlivé bloky z \mathbf{y}_0 mohou mít řídkou reprezentaci vzhledem k \mathbf{A}_a . Na základě bloků je tedy potřeba slovník patřičně upravit vzhledem k $\{\hat{\mathbf{q}}_a^k\}_{k=1}^N$ a tady přichází na řadu K-SVD algoritmus. Jako inicializační se bere DCT slovník.

V tento okamžik můžeme přistoupit k separaci. Zavedeme nové neznámé \mathbf{z}_c a \mathbf{z}_t , kde $\mathbf{z} = \mathbf{z}_c + \mathbf{z}_t$ a modifikujeme výraz $\|\mathbf{A}_a \mathbf{q}_a^k - \mathbf{R}_k \mathbf{z}\|_2^2$ na:

$$\begin{aligned} \left\| \mathbf{A}_{a} \mathbf{q}_{a}^{k} - \mathbf{R}_{k} \mathbf{z} \right\|_{2}^{2} &= \left\| \mathbf{A}_{c} \mathbf{q}_{c}^{k} + \mathbf{A}_{t} \mathbf{q}_{t}^{k} - \mathbf{R}_{k} \mathbf{z} \right\|_{2}^{2} \\ &= \left\| \mathbf{A}_{c} \mathbf{q}_{c}^{k} + \mathbf{A}_{t} \mathbf{q}_{t}^{k} - \mathbf{R}_{k} \mathbf{z}_{c} - \mathbf{R}_{k} \mathbf{z}_{t} \right\|_{2}^{2} \\ &\approx \left\| \mathbf{A}_{c} \mathbf{q}_{c}^{k} - \mathbf{R}_{k} \mathbf{z}_{c} \right\|_{2}^{2} + \left\| \mathbf{A}_{t} \mathbf{q}_{t}^{k} - \mathbf{R}_{k} \mathbf{z}_{t} \right\|_{2}^{2} \end{aligned}$$
(6.7)

Poslední úprava je možná, neboť předpokládáme, že chyba aproximace pro texturu a chyba aproximace pro trend jsou nekorelované. Tedy:

$$\left(\mathbf{A}_{c}\mathbf{q}_{c}^{k}-\mathbf{R}_{k}\mathbf{z}_{c}\right)^{\mathrm{T}}\left(\mathbf{A}_{t}\mathbf{q}_{t}^{k}-\mathbf{R}_{k}\mathbf{z}_{t}\right)\approx0.$$
(6.8)

S touto úpravou rovnice~(6.6) přechází do požadovaného tvaru pro lokální MCA:

$$\{\hat{\mathbf{y}}_{c}, \hat{\mathbf{y}}_{t}\} = \arg\min_{\mathbf{z}_{c}, \mathbf{z}_{t}} \lambda \|\mathbf{z}_{c} + \mathbf{z}_{t} - \mathbf{y}\|_{2}^{2} + \sum_{k=1}^{N} \left\|\mathbf{A}_{c}\mathbf{q}_{c}^{k} - \mathbf{R}_{k}\mathbf{z}_{c}\right\|_{2}^{2} + \sum_{k=1}^{N} \left\|\mathbf{A}_{t}\mathbf{q}_{t}^{k} - \mathbf{R}_{k}\mathbf{z}_{t}\right\|_{2}^{2}.$$
(6.9)

7 REALIZACE ALGORITMŮ V PROSTŘEDÍ MATLAB

V této kapitole se budeme zabývat realizací některých z výše uvedených algoritmů v prostředí MATLAB pro jednorozměrné a dvourozměrné signály. Vstupem do každé 1D aplikace je řídký signál ve zvolené bázi. Tato báze je použita i pro jeho rekonstrukci. Pro 2D aplikace k této volbě přibývá možnost vstupu i v podobě reálného obrazu.

Pro posouzení kvality rekonstrukce budeme používat kritérium PSNR (Peak Signal to Noise Ratio) – špičkový poměr signálu k šumu. Obecně PSNR vyjadřuje poměr mezi maximální hodnotou pixelu v nezašuměném obrazu a hodnotou šumu po rekonstrukci. Uvažujme obraz velikosti $M \times N$. Označme původní nezašuměný obraz I. Označne jako K obraz po rekonstrukci a maximální hodnotu pixelu v I jako MAX_I. Hodnotu šumu po rekonstrukci vyjádříme jako MSE (Mean Squared Error):

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{i=0}^{N-1} \|\mathbf{I}_{ij} - \mathbf{K}_{ij}\|_2^2$$
(7.1)

Pak PSNR:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$
(7.2)

V našem případě obraz nereprezentujeme vždy pomocí hodnot pixelu a proto budeme uvažovat maximální "hodnotu obrazu" ve smyslu:

$$MAX_I = \max(\mathbf{I}_{ij}) - \min(\mathbf{I}_{ij}).$$
(7.3)

Toto kritérium budeme používat jak v případě 2D signálu, tak u 1D signálu. Čím větší hodnota PSNR, tím je obraz kvalitnější.

7.1 1D Dekvantizace

Pro řešení problému dekvantizace (AQ) jsme použili nástoje lineárního programování obsažené přímo v prostředí MATLAB. Tomuto kroku předcházelo přepsání úlohy do vhodného tvaru:

$$\min_{\substack{\mathbf{u},\mathbf{v}\\ \mathbf{u},\mathbf{v}]=[\mathbf{w}]}} \underbrace{\mathbf{1}^{\mathrm{T}}\mathbf{u} + \mathbf{1}^{\mathrm{T}}\mathbf{v}}_{\mathbf{1}^{\mathrm{T}}\mathbf{w}}$$

a následné upravení původní podmínky:

$$-\frac{\Delta}{2}\mathbf{1} \le (\mathbf{y}_q - \mathbf{A}\mathbf{u} + \mathbf{A}\mathbf{v}) \le \frac{\Delta}{2}\mathbf{1}, \qquad \mathbf{u} \ge \mathbf{0}, \quad \mathbf{v} \ge 0$$

do tvaru:

$$-\frac{\Delta}{2}\mathbf{1} \le \left(\mathbf{y}_q - \underbrace{[\mathbf{A} - \mathbf{A}]}_{\tilde{\mathbf{A}}}[\mathbf{w}]\right) \le \frac{\Delta}{2}\mathbf{1}, \qquad \mathbf{w} \ge 0$$

poté přepsání do podoby jedné nerovnosti:

$$\begin{bmatrix} \tilde{\mathbf{A}} \\ -\tilde{\mathbf{A}} \end{bmatrix} \mathbf{w} \le \mathbf{1} \frac{\Delta}{2} - \begin{bmatrix} -\mathbf{y}_q \\ \mathbf{y}_q \end{bmatrix}$$

Takto upravenou úlohu jsme mohli již přímo realizovat pomocí funkce *linprog* v MATLABu. Pro simulaci 1D dekvantizace byl vytvořen program *signal_dequanting*, ve kterém jsou realizována všechna měření a pocházejí z něj veškěré uvedené výstupy. Po vygenerování řídkého signálu jsme na základě zvoleného počtu kvantizačních hladin tento signál kvantizovali. Mezní rozsah rozhodovacích úrovní byl stanoven o něco málo větší než je vzdálennost mezi největší a nejmenší hodnotou v signálu. V případě kdyje hodnota rovna přímo rozhodovací úrovni vyvstává otázka, kam tuto hodnotu přiřadit. Interval proto nebudeme uvažovat mezi rozhodovacími úrovněmi jako uzavřený, ale jako zprava uzavřený a zleva otevřený. Po rekonstukci je výsledný signál opět kvantizován. Byly srovnány oba výsledky kvantizace: kvantovaný originální signál a kvantovaný rekonstruovaný signál. Pokud si odpovídají, není PSNR_{kvant} tedy hodnota PSNR mezi oběma kvantováními, definována. Rovněž jsme porovnávali koeficienty signálu před kvantizací a po rekonstrukci.



Obr. 7.1: Vlevo zkvantovaný signál, vpravo porovnání rekonstrukce s originálem. PSNR = 39.13 dB. Použitá báze: DWT db10, délka signálu: n = 32, řídkost: k = 5



Obr. 7.2: Vlevo výsledek kvantování originálu i rekonstrukce je stejný, vpravo porovnání koeficientů. Výchozí signál je stejný jako v předcházejícím případě.

Na základě zavedení (AQ) předpokládáme, že kvantování obou signálů (originálu a rekonstrukce) bude stejné. Ve většině případů tomu tak opravdu je. Problém však nastává, je-li hodnota rekonstrukce signálu rovna přímo některé rozhodovací úrovni. V tomto případě je možný přeskok o kvantizační hladinu, ačkoliv rekonstrukce jsou si velmi blízké. Je to způsobeno právě tím, jak jsme si zavedli přiřazení kvantizační hladiny hodnotám signálu ležící na rozhodovací úrovni.



Obr. 7.3: Situace, kdy kvantování originálu neodpovídá kvantování rekontrukce; parametry signálu: báze DWT db10, n = 32, k = 5, PSNR = 31.4697, zelená – rozhodovací úroveň, černá – kvantizační hladiny.

Realizovali jsme několik měření. Pro jedno nastavení parametrů (délka signálu, řídkost, báze, případně použitý wavelet, hloubka dekompozice, počet kvatizačních hladin) bylo provedeno 100 opakování a byla vypočtena průměrná hodnota PSNR. Výsledky jsou uvedny v tabulkách, společně s celkovým časem výpočtu při daném počtu opakování.

	Báze DCT	délka signálu						
		řídkost: $k = 20$			řídkost: $k = 10$			
hl.	charakteristiky	32	64	128	32	64	128	
5	PSNR/dB	16.6892	18.5394	21.0647	18.2402	20.4347	23.0291	
	cas/s	9.637	30.472	205.512	8.916	30.858	213.172	
10	PSNR/dB	24.2462	26.4192	28.8002	26.1390	28.6940	31.8187	
	cas/s	9.619	33.944	232.112	9.400	33.046	231.273	
15	PSNR/dB	28.333	30.5899	32.7665	30.4036	32.6580	35.8102	
	cas/s	9.922	34.599	209.524	9.658	33.901	212.787	

Tab. 7.1: Porovnání hodnot PSNR v případě báze DCT na počtu kvantizačních hladin. Dle očekávání, s rostoucím počtem kvantizačních hladit přesnost rekonstrukce roste.

Jako první jsme realizovali měření pro signál řídký v bázi DCT. Pokud budeme hodnotit PSNR v závislosti právě na řídkosti signálu, dle očekávání lze z tabulek 7.1 a 7.2 vyčíst nepřímou úměru. Rozdíly nejsou až tak výrazné.

Báz	ze DCT, DWT	délka signálu						
řídkost: $k = 5$			DCT		haar $hd = 3$			
hl.	charakteristiky	32	64	128	32	64	128	
5	PSNR/dB	19.5270	21.7475	24.8936	20.9251	23.9846	26.4609	
	čas/s	9.283	32.926	224.358	6.460	7.687	11.031	
10	PSNR/dB	28.9325	30.9823	35.3584	27.3852	30.3160	33.1686	
	cas/s	9.387	33.872	233.411	6.885	8.105	10.781	
15	PSNR/dB	33.0356	35.8448	39.2762	32.4638	34.1218	36.6262	
	cas/s	9.737	33.119	220.638	6.816	8.057	10.247	

Tab. 7.2: Porovnání přesnosti rekonstrukce signálu povahy DCT a DWT, hloubka dekompozice hd = 3,

S rostoucí délkou signálu nám při dané řídkosti roste i PSNR, což opět není nic překvapivého, neboť poměr řídkosti a délky signálu se nám mění. Pokud bychom poměr mezi těmito dvěma parametry zachovali, budou hodnoty PSNR podobné (tabulka 7.5 ukazuje realizaci takového měření). Pokud bychom uvažovali dostatečně

velký statistický soubor, jednalo by se přímo o střední hodnotu.

Lze si povšimnout, že zpracování signálu řídkého v bázi DCT je pro větší délku signálu časově náročnou záležitostí.

Přesnost, s jakou jsou oba dva signály různé povahy zpracovány, lze považovat za podobnou. Pro určení signálu DCT je ale tato záležitost podstatně časově náročnější. Pokud bychom uvažovali signál délky 512, dostali bychom se u DCT na čas kolem tří minut pro jedno opakování.

Báze DWT		délka signálu						
ř	ídkost: $k = 5$	db3			db6			
hl.	charakteristiky	32	64	128	32	64	128	
5	PSNR/dB	21.1223	24.1408	27.7330	21.2840	24.3355	26.6970	
	čas/s	8.246	15.558	32.093	8.725	18.811	64.115	
10	PSNR/dB	29.3302	32.6618	35.5185	29.9348	31.8392	35.1645	
	čas/s	8.486	17.191	37.050	9.194	21.387	73.808	
15	PSNR/dB	33.8573	36.7605	39.4327	33.4376	36.4354	39.1232	
	čas/s	8.537	16.537	33.038	9.369	20.388	66.687	

Tab. 7.3: Porovnání PSNR pro různé druhy waveletů. Čím složitější wavelet uvažujeme, tím delší je doba jeho zpracování u delších signálů.

	Báze DWT		závislost na hloubce dekompozice db6								
ř	ídkost: $k = 5$	počet kvantovacích hladin									
hd	charakteristiky	5	10	15	20	25	30	35	40		
3	PSNR/dB	21.28	29.93	33.44	36.22	38.42	40.23	41.71	43.00		
	cas/s	8.72	9.19	9.37	9.42	8.91	9.04	9.08	9.00		
4	PSNR/dB	21.34	29.34	33.47	35.74	38.50	40.16	41.63	43.04		
	cas/s	8.76	9.16	9.45	9.14	8.98	9.09	8.93	8.98		
5	PSNR/dB	21.25	29.47	33.63	36.29	38.40	40.14	41.61	42.66		
	čas/s	8.89	9.05	9.54	9.81	9.09	9.10	9.18	9.04		

Tab. 7.4: Jak lze vidět, volba hloubky dekompozice hd nemá vliv na kvalitu a čas zpracování uvažovaného signálu. S rostoucím počtem kvantizačních hladin roste přesnost rekonstrukce. Tato závislost není lineární a po překročení učité hranice jsou si přesnosti podobné pro každou další. Tato hranice záleží na "rozkmitu" signálu.

Kvantování je ztrátový proces. Pokud však máme nějaké znalosti o povaze kvantovaného signálu a pokud bylo použito dostatečné množství kvantizačních hladin, je

	Báze DWT	db6	řídkost: délka signálu $/2$				
	hd = 3		de	élka sign	álu		
hl	charakteristiky	32	64	128	256	512	
10	PSNR/dB	25.1177	25.3914	25.5363	25.6209	25.7181	
cas/s		8.888	26.062	73.041	261.969	1043.630	

Tab. 7.5: pokud zachováme poměr délky signálu a řídkost, bude obraz zpracován se stejnou přesností pro různé délky signálu.

možné původní signál dostat zpět s dostatečnou přesností. Tato operace je však pro delší signály časově náročná. Časovou náročnost ovlivňuje především řídkost daného signálu.



Obr. 7.4: Grafické znázornění závislosti délky výpočtu na zvolené bázi signálu. Uvedené časové hodnoty jsou po realizaci 100 opakování pro danou délku signálu a řídkost.

7.2 2D Dekvantizace

Úloha pro 2D dekvantizaci byla řešena obdobně jako v jednorozměrném případě. Byl vygenerován obraz řídký ve zvolené bázi (v tomto případě jsme postupovali přes Kroneckerovův součin), který jsme následně "vektorizovali" a kvantizovali. Úlohu jsme tímto způsobem převedli přímo na jednorozměrný případ. Lze tedy očekávat podobné výsledky jako u 1D úlohy. Provedli jsme několik měření, ale již s menším počtem opakování. Pro simulaci 2D dekvantizace byl vytvořen program *sig-nal_dequanting_2D*, který umožnuje pracovat i s reálními obrazy. Uvažujeme černobílé obrazy, pokud vstupní obraz je barevný, převedeme jej. V případě reálného obrazu bylo sníženo původní rozlišení obrazu pro rychlejší zpracování. Pokud obraz přesahoval rozměr 16×16 byl rozdělen do bloků o této velikosti a každý blok byl dekvantizován zvlásť. Tyto dílčí části byly poté složeny zpět na své původní pozice. Opět bylo provedeno kvantování rekonstruovaného signálu a porovnání s nakvantovaným originálem.



Obr. 7.5: Zprava původní obraz, kvantovaný a rekonstru
ovaný obraz. Parametry obrázku: rozměr: 8×8 , řídkost: 20, báze: DWT db3, hd = 3, hl: 10; PSNR = 28.5759 dB

Na obrázku 7.5 a 7.6 lze vidět rekonstrukci řídkého obrazu, kde kvantovaní originálu a rekonstrukce si odpovídá. Rezidua mezi originálem a rekonstrukcí nepřekročí velikost šířky kvantizační hladiny $\frac{\Delta}{2}$ až na vyjímky – opět jako v jednorozměrném případě je to situace, kdy hodnota rekonstrukce odpovídá rozhodovací úrovni.



Obr. 7.6: Zprava rezidua rekonstrukce, kvantovaná rekonstrukce, rezidua po obou kvantovaních. Parametry obrázku: rozměr: 8×8 , řídkost: 20, báze: DWT db3, hd = 3, hl: 10; PSNR = 28.5759 dB, hodnoty po kvantování originálu i rekonstrukce si odpovídají

V případě zpracování reálného obrazu, pracujeme s větší škálou hodnot. Řídkost obrazu v nějaké bázi pouze předpokládáme a nemusí být dostatečná pro kvalitní

rekonstrukci. I již rekonstrukce původní škály barev může být komplikovaná. Zvýšením kvality by v případě reálného obrazu mohlo dojít použitím jiné báze něžli DWT a DCT, které uvádíme v práci např. použití curveletů, které dobře popisují hrany v obraze.



Obr. 7.7: Dekvantizace reálného obrazu Lena: původní obraz, kvantovaný obraz, výsledek rekonstrukce; použitá báze při zpracování: DCT, počet kvantizačních hladin: hl = 10, PSNR = 24.4373 dB, $PSNR_{OK} = 27.9244 dB$, $PSNR_{kvant} = 36.1681 dB$

Jak lze vidět v případě Leny, zahrneme-li ještě jeden hodnotící veličinu – výpočet PSNR mezi originálem a kvantovaným obrazem PSNR_{OK}, zjistíme, že rekonstrukce je v horším poměru kvality nežli samotné kvantování. Rekonsrukce nám poskytla pouze rozšíření škály barev blížící se původní.



Obr. 7.8: Dekvantizace reálného obrazu Lena: rezidua mezi originálem a rekonstrukcí, výsledek kvantovaní rekonstruovaného obrazu, rezidua mezi hodnotami po kvantování; použitá báze při zpracování: DCT, počet kvantizačních hladin: hl= 10, PSNR = 24.4373dB, $PSNR_{OK} = 27.9244dB$, $PSNR_{kvant} = 36.1681dB$

Při měření jsme se omezili na řídké obrazy. Počet opakování jsme stanovili na 50 nebo 30 opakování, výsledky jsou tedy orientační, pro vyšší vypovídající hodnotu by bylo potřeba většího statistického souboru.

hl = 5		opakování: 30			rozměr: 8×8			
řídkost: $k = 20$		počet kvantizačních hladin						
Báze	charakter.	5	10	15	20	25	30	
DWT	PSNR/dB	21.3981	28.2626	32.4855	35.1130	37.2465	38.5418	
– haar	cas/s	2.620	2.586	2.617	2.988	3.006	3.1843	
DWT	PSNR/dB	21.3755	27.9657	32.0169	35.1564	36.7853	38.8392	
-db3	čas/s	4.867	5.459	5.642	4.855	6.078	6.186	
DCT	PSNR/dB	21.6137	28.2752	32.1210	35.0442	37.3628	39.1376	
	čas/s	5.597	6.270	6.535	6.464	6.822	6.664	

Tab. 7.6: Hodnoty PSNR pro různé druhy bazí v závislosti na počtu kvantizačních hladin pro k = 20.

hl = 5		opakování: 30			rozměr: 8×8			
řídko	st: $k = 40$	počet kvantizačních hladin						
Báze	charakter.	5	10	15	20	25	30	
DWT	PSNR/dB	22.6257	28.5327	32.2123	34.7884	36.8122	39.3540	
– haar	cas/s	2.751	2.431	2.817	2.982	2.986	3.024	
DWT	PSNR/dB	22.2493	28.4222	32.2659	34.8443	36.2835	39.106	
– db3	čas/s	4.885	5.189	5.534	5.223	5.598	5.510	
DCT	PSNR/dB	22.8237	28.4985	32.2313	34.9334	36.5427	39.0604	
	cas/s	5.287	5.887	6.127	6.314	6.345	6.299	

Tab. 7.7: Hodnoty PSNR pro různé druhy bazí v závislosti na počtu kvantizačních hladin pro k = 40.

Z tabulky 7.11 si lze povšimnout, že pro wavelet haar nemá řídkost až takový vliv a kvalita rekonstrukce si je ve všech případech podobná, dokonce zvolíme-li hodnotu k=256, dostaneme pro hl=5, podobný výsledek.

	hl = 5	opakování: 30			rozměr: 16×16			
Bá	ze DWT – haar	počet kvantizačních hladin						
k	charakter.	5	10	15	20	25	30	
20	PSNR/dB	24.3978	31.6416	35.9016	39.0009	41.0841	42.5776	
	čas/s	7.131	7.641	7.939	8.157	8.731	8.736	
40	PSNR/dB	22.7155	29.9412	34.0080	36.8411	39.2549	40.8221	
	čas/s	7.374	7.822	7.952	8.028	8.061	8.693	
60	PSNR/dB	21.6040	29.3088	33.2332	35.8905	38.4310	40.2395	
	čas/s	7.2947	7.530	7.789	8.124	8.256	8.243	
80	PSNR/dB	21.0912	28.6971	32.5842	35.4175	37.8641	39.9962	
	čas/s	6.657	7.297	7.706	7.756	7.942	7.820	

Tab. 7.8: Hodnoty PSNR obrazu řídkého v báza DWT –
harr v závislosti na různých hodnotách řídkosti

Časová náročnost je pro větší rozměr obrazu dosti velká, jak lze vidět z tabulky 7.9. Pokud ovšem pracujeme s řídkými obrazy, můžeme se při vhodné volbě počtu dekvantizačních hladin dočkat přijatelné rekonstrukce kolem 40dB.

	hl = 5	opakování: 50				
		řídkost: $k = 4$	řídkost: $k = 8$			
Báze	charakteristika	rozměr: 8×8	rozměr: 16×16			
DWT	PSNR/dB	24.9635	27.9731			
db6	čas/s	13.027	356.001			
DCT	PSNR/dB	22.7026	28.4995			
	čas/s	11.814	732.735			

Tab. 7.9: Ukázka výpočtové časové náročnosti pro různé báze.

7.3 1D a 2D Dekompozice

K řešení problému dekompozice jsme využili nástojů konvexní optimalizace z toolboxu UNLocBox. Pro realizaci jsme si vybrali formulaci úlohy MCA ve tvaru 6.3 a řešili pomocí algoritmu forward backward (3.1), konkrétně jeho uzpůsobení pro úlohu separace. V našem případě budou mít gradienty tvar:

$$\nabla_{x_1} f(x_1, x_2) = \mathbf{A}_1^* \left(\mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2 - y \right)$$
$$\nabla_{x_2} f(x_1, x_2) = \mathbf{A}_2^* \left(\mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2 - y \right)$$

I v tomto případě jsem postupovali tak, že jsme úlohu řešili pro $[w] = [x_1, x_2]$ a výsledné řešení poté separovali. Pro simulaci dekvantizace byl vytvořen program decomposition_1D a decomposition_2D, ze kterého pocházejí všechny výstupy uvedené v této podkapitole.

Nejprve jsme vygenerovali dva signály(1D nebo 2D), kde byl každý řídký v nějaké bázi. Tyto signály jsme poté složili a obdrželi vstupní signál do programu. Po separaci jsme porovnali koeficienty obou komponent s jejich originály a vypočetli jejich PSNR. Tuto hodnotu jsme určovali pro pro jednotlivé komponenty a pro celkový signál, tedy složený vstupní a signál vzniklý složením obou komponent. V případě 2D jsme určovali pouze výsledné PSNR složených obrazů.

Tato metoda nebyla příliš úspěšná u jednorozměrných signálů, i když dokázala odhadnout ve většině případů, které koeficienty jsou nenulové, s jejich velikostí už nastával problém. Čím delší signál, tím přesnějších výsledků bylo dosaženo. U řídkých dvourozměrných signálů byla hodnota PSNR kolem 50dB.



Obr. 7.9: Porovnání vstupních komponenet a jejich odhadů a jejich koeficientů. PSNR = 27.58dB, u jednotlivých komponent nižší $PSNR_T = 14.82dB$, $PSNR_C = 25.52dB$, použitý báze: DCT, DWT-haar, délka signálu 256, $k_1 = k_2 = 4$

V případě reálných obrazů se nepodařilo obrazy úspěšně rozložit pomocí DWT a DCT. Báze nebyly schopny vhodně popsat hrany obrazu a výsledné složení se původnímu obrazu dosti vzdalovalo. Tento problém je možné vyřešit přidáním například curveletů [21] nebo contourletů [19] jako možnou volbu báze. V případě vygenerovaných řídkých obrazů bylo rozložení úspěšné.



Obr. 7.10: Zprava původní obraz, rezidum mezi složenými obrazy, obraz zpětně složený z výsledných komponent. Parametry obrázku: rozměr: 8×8 ; řídkost:k1= 4, k2 =6; báze: DCT, DWT db3; hd = 3, hl: 10; PSNR = 53.26 dB, PSNR_T = 37.56 dB, PSNR_C = 41.54 dB



Obr. 7.11: Výstupní komponenty. Parametry obrázku: rozměr: 8×8 ; řídkost:k1= 4,k2 =6; báze: DCT, DWT db3; hd = 3, hl: 10; PSNR = 53.26 dB, PSNR_T = 37.56 dB, PSNR_C = 41.54 dB

Pokud vstupní signál vytvoříme jako kombinaci dvou signálů řídkých v DCT bázi, může nastat možnost, že se nám pořadí komponent prohodí. Obdržíme tak malé hodnoty PSNR komponent napříč celkové příznivé PSNR.

opakování: 100		rozměr: 16×16			rozměr: 64×64			
$k_1 = k_1$	$_2 = 100$	PSNR/dB						
Ba	áze	PSNR	$\mathrm{PSNR}_{\mathrm{T}}$	$\mathrm{PSNR}_{\mathrm{C}}$	PSNR	$\mathrm{PSNR}_{\mathrm{T}}$	$\mathrm{PSNR}_{\mathrm{C}}$	
DCT	DWT haar	48.2398	22.3225	34.8582	53.1359	25.1831	24.6527	
DCT	DWT db5	47.4667	21.6632	32.7508	52.5123	24.6807	24.4485	
DCT	DCT	39.0393	19.4803	19.5354	54.3184	24.8519	24.7522	
DWT haar	DWT $db5$	49.1711	32.2690	31.2119	52.2325	24.7162	24.4767	
DWT haar	DWT haar	50.0790	32.0744	32.4143	51.6208	25.2758	25.1550	

Tab. 7.10: Přehled PSNR pro různé kombinace bází a rozměrů obrázku, hloubka dekompozice hd = 3.

Z tabulky 7.3 můžeme vyčíst, že pokud jsou obě komponenty velmi řídké, je rekonstrukce každé z nich velice přesná. Se zvyšujícím se k hodnota PSNR jednotlivých komponent klesá.

opakování: 20		rozměr: 16×16			rozměr: 64×64		
PSNR/dB		$\check{\mathrm{r}\mathrm{i}\mathrm{d}\mathrm{kost}}:\mathbf{k}_1=\mathbf{k}_2=4$			$\check{\mathrm{r}}\mathrm{i}dkost:\mathrm{k}_1=\mathrm{k}_2=20$		
Báze		PSNR	PSNR _T	PSNR _C	PSNR	PSNR _T	$\mathrm{PSNR}_{\mathrm{C}}$
DCT	DWT haar	56.4630	55.7201	52.1044	48.1182	27.5012	20.5538
DCT	DWT db5	55.4299	54.5781	52.1415	48.1137	27.6926	19.8634
DCT	DCT	59.6752	17.8272	18.0183	43.6224	18.7198	18.5617
DWT haar	DWT db5	55.4329	44.5807	43.8894	47.9631	26.8392	25.5116
DCT	DWT db3	55.7495	55.5185	52.2758	48.6595	28.2034	20.5438

Tab. 7.11: Přehled PSNR pro různé kombinace bází, řídkosti a rozměrů obrázku, hloubka dekompozice hd = 3.

7.4 1D a 2D Odšumování signálu

Úlohu odšumování obrazu jsme opět realizovali pomocí nástrojů konvexní optimalizace obsažené v toolboxu UNLocBox. Tentokráte jsme využili formulace algoritmu Douglas-Rachford a aplikovali jej na úlohu (BPDN).

Pro simulaci odšumování obrazu byl vytvořen program *signal_denoising* a *sig-nal_denoising_2D*, ze kterého pocházejí všechny výstupy uvedené v této podkapitole.

Nejprve jsme vygenerovali signál řídký ve zvolené bázi a přidali k němu aditivní

Gaussův šum. Šum přidáváme pomocí hodnoty SNR, čím je hodnota SNR menší, tím větší je šum. Takto jsme obrdželi vstupní signál do našeho algoritmu.



Obr. 7.12: Zašuměný signál a rekonstrukce v porovnání s originálem. Parametry signálu: délka signálu: 64, řídkost: k = 10, použitá báze DCT, vstupní poměr šumu: snr = 1. PSNR = 23.2799 dB, PSNR_{sr} = 18.7352 dB.

Na první pohled se algoritmus nezdá příliš přesný, abychom ověřili, zda byl nějaký šum odstaněň, je pro orientaci vypočítána i hodnota PSNR_{sr}, která nám udává PSNR mezi zašuměným signálem a originálem. Pokud se nedostaneme pod tuto hodnotu, šum být v nějaké míře odstraněn.



Obr. 7.13: Porovnání koeficientů původního signálu s rekonstrukcí, porovnání odstraněného a přidaného šumu. Parametry signálu: délka signálu: 64, řídkost: k = 10, použitá báze DCT, vstupní poměr šumu: snr = 1. PSNR = 23.2799 dB, PSNR_{sr} = 18.7352 dB.

Intuitivně vyvstává otázka, zda v případě koeficientů z obrázku 7.13 za předpokladu pevné řídkosti k, bychom nemohli ponechat pouze k nejvyšších hodnot a zbylé koeficienty rekonstrukce položit nule. Hodnotu k ale ve většině případů neznáme a není zaručeno, že největší hodnoty u rekonstruovaného signálu odpovídají právě těm původním.

opakování: 100		řídkost:					
délka signálu: 128		2	10	20	30	40	60
DWT	PSNR	20.9190	26.9974	24.3953	22.8983	21.9390	20.5084
db3	$\mathrm{PSNR}_{\mathrm{sr}}$	16.5010	22.0863	20.5409	19.7861	19.3222	18.7097
DWT	PSNR	30.9648	26.7267	24.2177	22.7466	21.9537	20.4242
haar	$\mathrm{PSNR}_{\mathrm{sr}}$	24.8121	21.8114	20.2816	19.6305	19.2425	18.6377
DCT	PSNR	21.0441	22.8083	22.0482	21.2940	20.9077	20.0346
	$\mathrm{PSNR}_{\mathrm{sr}}$	14.8315	17.9599	18.1113	18.1944	18.3156	18.2730

Tab. 7.12: Parametry signálu: poměr vstupního šumu: snr = 1, hloubka dekompozice hd = 3. Ve všech případech jde vidět mírné zlepšení kvality, ačkoliv šum nebyl plně odstraněn.

Jak jsme již zmínili v podkapitole 3.1.2, jelikož Douglas-Rachford provádí v každé iteraci dva proximální kroky, je podstatně časově náročnější na výpočet. Pro zjednodušení jsme se tedy omezili na signály, které potřebují na svou rekonstrukci maximálně 200 iterací. Ačkoliv je dle 3.9 konvergence zajištěna, nemusí být dostatečně "rychlá", aby ji bylo možné zpracovat v rozumném čase.



Obr. 7.14: Přidání šumu do obrazu. Parametry obrazu: rozměr: 16, řídkost: k = 4, použitá báze DWT haar, vstupní poměr šumu: snr = 10, hloubka dekompozice: hd = 3; PSNR = 41.781 dB, PSNR_{sr} = 34.921 dB.



Obr. 7.15: Jako rezidua jsme označili rozdíl hodnot mezi vstupním signálem a rekonstrukcí, odstraněný šum je hodnota rezidua odečtena od přidaného šumu. Parametry obrazu: rozměr: 16, řídkost: k = 4, použitá báze DWT haar, vstupní poměr šumu: snr = 10, hloubka dekompozice: hd = 3; PSNR = 41.781 dB, PSNR_{sr} = 34.921 dB.

V případě větších obrázků zpracováváme opět po blocích. U zpracování reálného obrazu opět narážíme na problém nevhodné báze a tak může nastat i situace, kdy zašuměný signál bude lepší kvality nežli naše rekonstrukce, jak lze vidět na obr 7.16. V tomto případě je původní obrázek v rekonstrukci velice málo rozeznatelný, o čemž svědčí i hodnota PSNR, která je nižší jak PSNR_{sr}.



Obr. 7.16: Ukázka rekonstrukce reálného obrazu pomocí báze DWT haar, vstupní poměr šumu: snr = 5, hloubka dekompozice: hd = 3; PSNR = 11.8260 dB, PSNR_{sr} = 21.0780 dB.

ZÁVĚR

Cílem této diplomové práce bylo nastudovat základní principy a algoritmy pro hledání řídké reprezentace dat a provést praktické ověření vybraných metod na reálných i silumovaných datech. V úvodu práce jsme se seznámili se základními pojmy této problematiky a uvedli si princip ℓ_1 -relaxace, který nás provázel celou prací.

Pro řešení aplikační úlohy odstranění šumu jsme si pro realizaci vybrali Douglas-Rachford algoritmus, který neklade žadné speciální požadavky na minimalizovanou funkci a zdá se tak univerzálním nástrojem. Jedná se však o úlohu náročnou na výpočet díky dvěma proximity krokům v každé iteraci. Výsledky nejsou příliš přesné. Ačkoliv v zašumělém signálu dojde ke zlepšení oproti původní hodnotě šumu, stále nám podstatná část šumu zůstane v obraze přítomna. U reálných obrazů obdržíme i horší kvalitu nežli za působení samotného šumu. Toto je však způsobeno tím, ze vstupní obraz není dostatečně řídký v zadané bázi, a tak může obraz místy špatně reprezentovat.

Úspěšnější aplikací byla realizace úlohy dekvantizace pomocí myšlenky Dantzig selektoru. Formulovali jsme úlohu do podoby lineárního programování a provedli několik měření. Ukázalo se, že tato rekonstrukce může být dosti přesná, ale pro některé druhy signálu z časového hlediska nevýhodná. Pro každou silumaci jsme prováděli i zpětné kvantování, abychom ověřili přesnost rekonstrukce, tj. zda všechny hodnoty náleží svým příslušným kvantizačním hladinám. V tomto případě nastala výjimka, kdy si tyto hodnoty neodpovídaly. Jednalo se o případ, kdy hodnota rekonstrukce připadla přímo na rozhodovací úroveň, a tak byla dle našeho zavedení kvantování přesenesena na vedlejší kvantizační hladinu. I přesto přinášela tato metoda dobré výsledky.

Poslední aplikací byla dekompozice signálu. V tomto případě byly výsledky podstatně slabší. Nejprve jsme vygenerovali dva řídké signály a složili jsme je v jeden. Poté jsme se pokoušeli se najít původní komponenty před složením. U jednorozměrných signálů v několika případech došlo k prohození komponent. Rekonstrukce za této situace neproběhla úspěšně. Jednalo se o případy, kdy jsme zvolili stejné báze u obou vstupních signálů. Pro dvourozměrný řídký signál naopak měření vycházela ve vysokých hodnotách PSNR. V případě reálného obrazu byla dekompozice nekvalitní. Zpětným složením jsme mnohdy neobdrželi ani příliš podobný obraz. Velký podíl na tom má skutečnost, že aproximace komponent probíhala v kombinaci s DCT nebo DWT bázemi. Neměli jsme žádnou bázi, která by byla schopna dobře popsat hrany, které se v obraze nacházely. Tento problém by případně mohlo vyřešit použití curveletové transformace. Na práci by se takto dalo navázat.

Další zajímavou variantou by mohlo být použití dekompozice v případě kvantování. Nevýhodou tohoto způsobu zpracování je větší časová náročnost oproti běžným konvenčním metodám.

LITERATURA

- Elad M. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, ISBN 978-1-4419-7010-7, 2010
- [2] Hrbáček R., Rajmic P., Veselý V., Špiřík J., Řídké reprezentace signálů: úvod do problematiky. Elektrorevue, 2011
- [3] Tebbens J.D., Hnětynková I., Plešinger M., Strakoš Z., Tichý P., Analýza metod pro maticové výpočty, Základní metody. MATFYZPRESS, ISBN 978-80-7378-201-6, 2012
- [4] Rajmic P., Průša Z., Discrete Wavelet Transform of Finite Signals: Detailed Study of the Algorithm. International Journal of Wavelets, Multiresolution and Information Processing, 2013
- [5] Fornasier M., Rauhut H., Handbook of Mathematical Methods in Imaging. Springer, ISBN 978-0-387-92920-0, 2011
- Beck A., Teboulle M., A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. SIAM J. IMAGING SCIENCES, vol.2, no.1, pp.183-202, 2009
- [7] Olshausen B. A., Field D. J., Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, vol.381, pp.607-609, 1996
- [8] P. Combettes a J. Pesquet, Fixed-Point Algorithms for Inverse Problems in Science and Engineering. kapitola Proximal Splitting Methods in Signal Processing, Springer, pp.185-212, 2011
- [9] Combettes P., Pesquet J., A douglas-rachford splitting approach to nonsmooth convex variational signal recovery. Selected Topics in Signal Processing, IEEE Journal of, vol.1, no.4, pp.564-574, 2007
- [10] Gill P.R, Wang A., Molnar A., The In-Crowd Algorithm for Fast Basis Pursuit Denoising. Signal Processing, IEEE Transactions on , vol.59, no.10, pp.4595-4605, 2011
- Schmidt M., Least Squares Optimization with L1-Norm Regularization. Project Report, CS542B, 2005.
- [12] Combettes P.L., Wajs V.R., Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul., vol.4, pp.1168-1200, 2005

- [13] Hundsdorfer W.H., Verwer J.G., Stability and Convergence of the Peaceman-Rachford ADI Method for Initial-Boundary Value Problems. Mathematics of Computation, vol.53, no.187, pp.81-101, 1989
- [14] Candes E., Tao T., The Dantzig selector: statistical estimation when p is much larger than n. The Annals of Statistics, vol.35, no.6, pp.2313-2351, 2007
- [15] Liang L., Yongcheng L., Qing L., Fast Algorithms to Solve the Dantzig Selector. Control and Automation (ICCA), 2013 10th IEEE International Conference on, pp.1544 - 1549, 2013
- [16] James G., Radchenko P., Lv J., DASSO: connections between the Dantzig selector and lasso. Journal of Royal Statistical Society Series B, vol.71, pp.127–142, 2009
- [17] Asif M. S., Romberg J., On the LASSO and Dantzig selector equivalence Information Sciences and Systems (CISS), 2010 44th Annual Conference on, pp.1-6, 2010
- [18] Mayer Y., Averbuch A., Coifman R., Multilayered image representation: Application to image compression IEEE Trans. on Image Proceesing, vol.11, no.9, 2002
- [19] Fadili M.J., Starck J-L, Elad M., Donoho D.L., MCALab: Reproducible Research in Signal and Image Decomposition and Inpairing Computing in Science & Engineering, vol.12, no.1, pp.43-63, 2010
- [20] Fadili M.J., Starck J-L, Bobin J., Moudden Y., Image Decomposition and Separation Using Sparse Representations: An Overview Proceedings of the IEEE, vol.98, no.6, pp.983-994, 2010
- [21] Candes E., Demanet L., Donoho D., Ying L., Fast Discreate Curvelet Transforms Multiscale Modeling & Simulation, vol.5, no.3, pp.861-899, 2006

SEZNAM PŘÍLOH

A První příloha	63
A.1 Přiložené programy – popis funkcí $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	63
A.1.1 2D úlohy	64
A.1.2 Možnosti vyhození chyby	64
A.1.3 Ukázka uživatelského prostředí	64
B Druhá příloha	67
B.1 Trénované slovníky	67

A PRVNÍ PŘÍLOHA

A.1 Přiložené programy – popis funkcí

Všechny uvedené aplikace byly realizovány v prostředí MATLAB pomocí uživatelkého prostředí GUI. V každé složce nalezneme dva soubory s názvem aplikace s koncovkou .m, vybereme ten bez přípony imf a spustíme. Po jeho kompilaci je nám objeví uživatelké prostředí, do kterého můžeme zadávat a měnit vstupní parametry. Žádná ze složek nepotřebuje další speciální toolbox, vše potřebé je obsaženo uvnitř složky.

Při vypracovávání aplikací jsem použili některé funkce z toolboxu pro konvexní optimalizaci UNLocBoX dostupné z : unlockbox.sourceforge.net/index.php

- 1. *solve_bpdn.m* funkce pro výpočet úlohy BPDN
- 2. test_gamma.m posuzuje korektnost parametru Γ
- 3. soft_thresholdb.m měkké prahování
- 4. soft_threshold.m měkké prahování
- 5. $prox_l1.m$ proximalní operátor pro řešení proximálního kroku
- 6. $proj_b2.m$ projekce na B2
- 7. $convergence_test.m$ testuje konvergenci dané metody
- 8. test_weights.m
- 9. test_multigroup.m

Dále využíváme pomocné funkce od autorů Zdeněk Průša a Pavel Rajmic funkce:

- 1. *w_filters.m* vyvoří si paramerty pro fci *dwtmtrix*
- 2. dwtmtrix– výpočet bázové funkce pro konkrétní waveletou transformaci

Pro realizaci jednotlivých úloh byly vytvořeny následující funkce:

- 1. *signal_dequanting* Na základě vstupním parametrů vygeneruje řídký signál, provede kvantování a následně dekvantizaci. Vypočte PSNR – počítáno mezi vstupním a rekonstruovaným signálem, vykreslí koeficienty původního a rekonstruovaného signálu.
- signal_dequanting_2D Na základě vstupním parametrů vygeneruje řídký obraz, provede kvantování a následně dekvantizaci. Vypočte PSNR, PSNR_OK počítáno mezi vstupním signálem a kvantovaným, PSNR_sr počítáno mezi kvantovaným signálem a rekonstrukcí.
- 3. signal_denoising Na základě vstupním parametrů vygeneruje řídký signál, přidá šum do signálu a provede odstranění šumu. Vypočte PSNR, PSNR_sr
 v tomto případě mezi zašuměným signálem a puvodním signálem

- 4. *signal_denoising_2D* Analogicky jako v 1D případě.
- signal_decomposition Vygeneruje dva řídké signály ve zvolených bází a složí je. Takto vzniklý signál je opět rozložen na dvě komponenty signal_tex a signal_car. Vypočte PSNR a PSNR mezi jednotlivými složkami: PSNR_T, PSNR_C.
- signal_decomposition_2D V případě volby náhodného signálu vygeneruje dva řídké obrazy, které posleze složí. Takto vzniklý signál je algoritmem rozložen opět na dva. Vypočte PSNR.

A.1.1 2D úlohy

V případě 2D úloh, pokud nastavíme rozměr obrázku větší jak 16×16 je obrázek rozdělen do bloků a ty jsou zpracovány samostatně. Velikost bloku je stanovena na 16×16 , možné změnit uvnitř příslusné funkce – končí *__impl.m*, změnou *velikost_deleni*. V případě reálného obrazu je jeho velikost upravena na 32×32 opět možno změnit v příslušné funkci.

A.1.2 Možnosti vyhození chyby

Vyhození chyby, ukončení výpočtu bez dokončení celého procesu: Vstupní signál nevyhověl některé z podmínek uvedené ve funkcích z UNLocBoX - překročen nastavený počet iterací, nevyhovující rychlost konvergence... Aplikaci je možné dál užívat po vygenerování dalšího signálu.

A.1.3 Ukázka uživatelského prostředí

Po spustění aplikace se vstupní i výstupní hodnoty vypíši do CommandWindow. Vypíší se i hodnoty, které jsme nevolili. Například pracujeme s bází DCT a vypíše se nám wavelet "haar". V tomto případě wavelet samozřejmě nebyl použit, jedná se o defaulní hodnotu při možném nevýběru báze DWT. Pokud je použita funkce z UNLocBoX, je nastaveno vypsání všech iterací, toto je možnost změnit přes *verbose* – položit nule, rovněž lze změnit maximální počet povolených iterací změnou hodnot u *paramfb.maxit*. Obě dvě hodnoty se nachazejí ve funkcích s příponou _*impl.m* není je třeba hledat v konkrétních pomocných funkcí z UNLocBoX.

V jednotlivých aplikacích je možné použít nástroje přiblížení, oddálení, posun. Aplikace byly vytvořeny v prostředí MATLAB R2011b, pro novější verze MATLAB může dojít ke zalomení aplikace i uprostřed grafu, pro tento případ je nutné přenastavení v souboru s příponou *.fig.*



Obr. A.1: Uživatelské prostředí pro úlohu 1D kvantizace.



Obr. A.2: Uživatelské prostředí pro úlohu 2D kvantizace.

B DRUHÁ PŘÍLOHA

B.1 Trénované slovníky

Označení B.1. Označme **Y** jako matici, jejiž sloupce tvoří jednotlivé vektory \mathbf{y}_i , a **X** jako matici, jejiž sloupce tvoří jednotlivé vektory \mathbf{x}_i .

Metoda optimálních směrů – MOD

V případě metody optimálních směrů se jedná o složenou minimalizační úlohu vnitřní a vnější minimalizace. Vnitřní minimalizace je zaměřena na počet nenulových prvků v reprezentaci jednotlivých \mathbf{x}_i , kde \mathbf{A} je pevně dáno. Vnější minimalizace pracuje se samotným slovníkem \mathbf{A} .

Metoda pracuje následovně: V k-tém kroku použijeme slovník z předchozího kroku \mathbf{A}_{k-1} a s jeho využitím obdržíme matici \mathbf{X}_k . Pomocí metody nejmenších čtverců hledáme matici \mathbf{A}_k :

$$\mathbf{A}_{k} = \arg\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}_{k}\|_{F}^{2} = \mathbf{Y}\mathbf{X}_{k}^{T}(\mathbf{X}_{k}\mathbf{X}_{k}^{T})^{-1} = \mathbf{Y}\mathbf{X}_{k}^{+}.$$
 (B.1)

Algoritmus je ukončen po dosažení požadované přesnosti.

Metoda K-SVD

Zatímco metoda MOD hledá matici **A** jako celek a v každé iteraci ji zpřesňuje, metoda K-SVD zpracovává jednotlivé atomy zvlášť. Zafixujeme všechny atomy kromě \mathbf{a}_{j_0} , který aktualizujeme pomocí přenásobení koeficienty z **X**. Celý algortimus pracuje následovně [1]:

K-SVD Algoritmus

Inicializace: k = 0 a

- Inicializace slovníku:
sestavme matici $\mathbf{A}_0\in\mathbb{R}^{n\times m}$ pomocí náhodných hodnot nebo pomocí náhodně
 mvybraných příkladů
- Normování: normování sloupců matice ${\bf A}$

Iterace: v každé iteraci zvyško jedna a proveď

• Fáze řídkého kódování: Aproximací řešení úlohy

$$\hat{\mathbf{x}}_{i} = \arg\min_{\mathbf{x}} \|\mathbf{y}_{i} - \mathbf{A}_{k-1}\mathbf{x}\|_{2}^{2} \quad \text{vzhledem k} \quad \|\mathbf{x}\|_{0} \le k_{0} \quad (B.2)$$

obdržíme řídkou reprezentaci $\mathbf{\hat{x}}_i$ pro $1 \leq i \leq M$. Přeskládáme do matice \mathbf{X}_k .

- K-SVD aktualizace slovníku: Použitím následujícího postupu k aktualizaci sloupců slovníku obdržíme \mathbf{A}_k : Opkakujem pro $j_0 = 1, 2, \ldots, m$
 - $-\,$ zaveďení množinu vzorů, kde je použit atom \mathbf{a}_{j_0} jako

$$\Omega_{j_0} = \{ i \mid 1 \le i \le M, \mathbf{X}_k \, [j_0, i] \ne 0 \}$$
(B.3)

– výpočet reziduální matice

$$\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j \neq j_0} \mathbf{a}_j \mathbf{x}_j^T, \tag{B.4}$$

kde \mathbf{x}^j jsou j-té řádky v matici \mathbf{X}_k

- restrikce \mathbf{E}_{j_0} výběrem pouze sloup
ců korespondující s Ω_{j_0} odbržení $\mathbf{E}_{j_0}^R$
- SVD dekompozice $\mathbf{E}_{j_0}^R = \mathbf{U} \Delta \mathbf{V}^T$, aktualizece atomu slovníku $\mathbf{a}_{j_0} = \mathbf{u}_1$, reprezentace $\mathbf{x}_{j_0}^R = \mathbf{\Delta} [1, 1] \cdot \mathbf{v}_1$
- Zastavující kritérium: ukončíme pokud je $\|\mathbf{Y} \mathbf{A}_k \mathbf{X}_k\|_F^2$ dostatečně malé, v opačném případě následuje další iterace

Výstup: slovník A_k

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

DCT	Diskrétní kosinová transformace
DWT	Diskrétní waveletová transformace
MOD	Metoda optimálních směrů
NSP	Null Space Property
RIP	Restricted Isometry Property
BPDN	Basis Pursuit Denoising
LASSO	Least Absolute Selection and Shrinkage Operator
QP	Quadratic Programming
POSC	Projection Onto Convex Sets
DS	Dantzig Selector
ROP	Restricted Orthogonality Property
LADMM	Linearized Alternating Direction Method
DASSO	Dantzig Selector with Sequential Optimization
ROP	Restricted Orthogonality Property
MCA	Morfologická analýza komponent
MAP	Maximum a posteriori probability
1D	jednodimenzionální úloha
2D	dvoudimenzionální úloha
PSNR	Peak Signal to Noise Ratio
SNR	Signal to noise ratio

$\operatorname{supp}(\mathbf{x})$	nosič vektoru
A	matice A rozměru $m \times n$
x	vektor rozměru \boldsymbol{n}
У	vektor rozměru \boldsymbol{n}
k	řídkost matice
Σ_k	množina všech k -řídkých vektorů
σ_k	chyba nejlepší aproximace
\mathbf{A}^*	hermitovsky sdružená matice
\mathbf{A}^{T}	hermitovsky sdružená matice
\hat{a}_{ji}	číslo komplexně sdružené k a_{ji}
I	jednotková matice
ϵ	uvažovaná odchylka od přesného řešení
$\operatorname{spark}(\mathbf{A})$	spark matice \mathbf{A}
$\operatorname{rank}(\mathbf{A})$	hodnost matice \mathbf{A}
$\mu(\mathbf{A})$	koherence matice \mathbf{A}
F	matice Fourierovy transformace
h	filtr typu dolní propust
g	filtr typu horní propust
ĥ	rekonstrukce filtru typu dolní propust
ĝ	rekonstrukce filtru typu horní propust
W	sloupce matice tvoří wavelety
Н	řádky matice tvořeny filtry typu dolní propust
G	řádky matice tvořeny filtry typu horní propust
J	hloubka dekompozice
$\downarrow 2$	dyadické podvzorkování

a	aproximativní waveletové koeficienty
d	detailní waveletové koeficienty
$\ker \mathbf{A}$	jádro matice \mathbf{A}
η	vektory jádra matice \mathbf{A}
γ	konstanta NSP
δ	konstanta zeslabené izometrie
P_C	projekční operátor
C_i	konvexní množiny
$\operatorname{dom} f$	definiční obor funkce f
Γ	třída zdola polospojitých konvexních funkcí
d_C	vzdálenost funkce od neprázdné množiny
$\operatorname{prox} f_i$	proximity operátor funkce f_i
\mathbb{R}	množina reálných čísel
\mathbb{C}	množina komplexních čísel
λ	relaxační parametr
e	Gaussovský bílý šum s rozdělením $e_i \sim \mathcal{N}(0, \sigma^2)$
heta	konstanta zeslabené ortogonality
Ŷ	odhad vektoru ${\bf x}$
heta	konstanta zeslabené ortogonality
Δ	šířka kvantizační hladiny
\mathbf{R}_k	operátor přístupu ke $k\text{-t}\acute{\mathrm{e}}\mathrm{mu}$ prvku