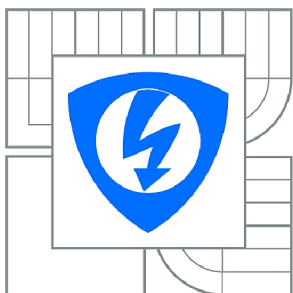




**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

# **APLIKACE METOD UČENÍ SLOVNÍKU PRO AUDIO INPAINTING**

APPLICATIONS OF DICTIONARY LEARNING METHODS FOR AUDIO INPAINTING

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. ROMAN OZDOBINSKI**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÁCLAV MACH**

BRNO 2014



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Roman Ozdobinski

**ID:** 125577

**Ročník:** 2

**Akademický rok:** 2013/2014

**NÁZEV TÉMATU:**

## Aplikace metod učení slovníku pro Audio Inpainting

### POKYNY PRO VYPRACOVÁNÍ:

Pro metody učení transformačního slovníku využijte algoritmus pro segmentaci stacionárních úseků vzorového signálu a prověřte jeho efektivitu ve srovnání s klasickým výběrem trénovacích atomů. Dále využijte pro porovnání efektivitu učení slovníků jeden z algoritmů pro adaptaci inkoherentního slovníku. Tyto techniky učení slovníků porovnejte pro hudební nahrávky různých žánrů (databáze GTZAN) a vyhodnoťte optimální parametry učení slovníků pro jednotlivé žánry. Připravte testovací skripty pro hromadné testy a vyhodnoťte úspěšnost rekonstrukce.

### DOPORUČENÁ LITERATURA:

[1] BARCHIESI, Daniele. Sparse Approximation and Dictionary Learning with Applications to Audio Signals. Londýn, 2013. Disertační práce. Queen Mary, University of London.

[2] Aharon, M.; Elad, M.; Bruckstein, A. M.: K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. IEEE Transactions on Signal Processing, ročník 54, 2006: s. 4311–4322.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 28.5.2014

**Vedoucí práce:** Ing. Václav Mach

**Konzultanti diplomové práce:**

**doc. Ing. Jiří Mišurec, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato diplomová práce se zabývá metodami učení slovníku pro doplňování chybějících úseků ve zvukovém signálu. Jsou zde teoreticky rozebrány a prakticky využity algoritmy K-SVD a INK-SVD k učení slovníku. Tyto slovníky jsou následně aplikovány na rekonstrukce audio signálů metodou OMP (Orthogonal Matching Pursuit). Dále je zde navrhnutý algoritmus pro výběr stacionárních segmentů a jejich následné použití jako trénovací data pro K-SVD a INK-SVD. V praktické části práce je pozorována účinnost při výběru do trénovací množiny z celého signálu v porovnání s použitím algoritmu pro stacionární segmentaci. Byl také zkoumán vliv snížení vzájemné koherence na kvalitu rekonstrukce pro inkoherentní slovník. Vytvořenými skripty pro hromadné testování v prostředí Matlab je provedeno vzájemné porovnání uvedených metod na žánrově odlišených skladbách.

## **KLÍČOVÁ SLOVA**

K-SVD, INK-SVD, DCT, inkoherentní, stacionární, OMP, učení slovníku, inpainting

## **ABSTRACT**

This diploma thesis discusses methods of dictionary learning to inpaint missing sections in the audio signal. There was theoretically analyzed and practically used algorithms K-SVD and INK-SVD for dictionary learning. These dictionaries have been applied to the reconstruction of audio signals using OMP (Orthogonal Matching Pursuit). Furthermore, there was proposed an algorithm for selecting the stationary segments and their subsequent use as training data for K-SVD and INK-SVD. In the practical part of thesis have been observed efficiency with training set selection from whole signal compared with algorithm for stationary segmentation used. The influence of mutual coherence on the quality of reconstruction with incoherent dictionary was also studied. With created scripts for multiple testing in Matlab, there was performed comparison of these methods on genre distinct songs.

## **KEYWORDS**

K-SVD, INK-SVD, DCT, incoherent, stationary, OMP, dictionary learning, inpainting

OZDOBINSKI, Roman *Aplikace metod učení slovníku pro Audio Inpainting*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 56 s. Vedoucí práce byl Ing. Václav Mach

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Aplikace metod učení slovníku pro Audio Inpainting“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Václavu Machovi za odborné vedení, přínosné konzultace, cenné rady a podnětné návrhy při realizaci práce. Dále děkuji Mgr. Pavlu Rajmicovi, Ph.D. za dodatečné rady a neocenitelné zkušenosti získané ze zahraničních projektů a užítované v diplomové práci. V neposlední řadě děkuji své rodině za podporu a trpělivost.

Brno .....

.....

(podpis autora)



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Technická 3058/10, CZ-61600 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsany 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

Úvod	11
<b>1 Řídké reprezentace signálu</b>	<b>12</b>
1.1 Definice základních pojmů	12
1.2 Řídká řešení a podmínky	13
<b>2 Rekonstrukce zvukového signálu</b>	<b>15</b>
2.1 Výběr poškozených vzorků	15
2.2 Aproximační algoritmy	16
2.2.1 Algoritmus OMP	17
2.3 Hodnocení kvality rekonstrukce	18
2.3.1 Výpočet SNR	18
2.3.2 PEMO-Q	18
<b>3 Slovníky</b>	<b>19</b>
3.1 Statické slovníky	19
3.1.1 DCT slovník	19
3.2 Naučené slovníky	20
3.2.1 K-SVD	20
3.2.2 Inkoherentní K-SVD	20
3.3 Vizualizace slovníků	20
<b>4 Algoritmus K-SVD</b>	<b>22</b>
4.1 Kvantizace vektorů a K-means	22
4.2 Popis algoritmu	22
4.2.1 Použití K-means	22
4.2.2 Použití SVD	23
4.3 Optimální parametry pro K-SVD	26
<b>5 Učení inkoherentního slovníku</b>	<b>27</b>
5.1 Vzájemná koherence	27
5.2 Inkoherentní K-SVD	27
5.3 Algoritmus INK-SVD	28
5.3.1 Dekorelace dvou atomů	28
5.3.2 Dekorelace celého slovníku	30
5.3.3 Vizualizace odlišnosti učených slovníků	31

<b>6</b>	<b>Metody sběru dat</b>	<b>32</b>
6.1	Výběr z celého signálu . . . . .	32
6.2	Výběr z lokální oblasti signálu . . . . .	32
6.2.1	Segmentace na stacionární úseky . . . . .	33
6.2.2	Aplikace segmentace pro výběr do trénovací množiny . . . . .	33
<b>7</b>	<b>Implementace a aplikace</b>	<b>36</b>
7.1	Popis funkcí . . . . .	36
7.2	Implementace PEMO-Q . . . . .	37
7.3	Knihovna hudebních signálů . . . . .	37
<b>8</b>	<b>Výsledky a porovnání</b>	<b>39</b>
8.1	Hodnocení metod sběru dat . . . . .	39
8.1.1	Aplikované sady děr . . . . .	39
8.1.2	Experiment pro trénovací data z celé skladby . . . . .	41
8.1.3	Experiment pro trénovací data ze stacionárních segmentů . . . . .	42
8.2	Srovnání algoritmů pro učení slovníků . . . . .	43
8.3	Hodnocení na základě PEMO-Q . . . . .	43
8.4	Poslechové testy . . . . .	44
<b>9</b>	<b>Závěr</b>	<b>45</b>
	<b>Literatura</b>	<b>46</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>48</b>
	<b>Seznam příloh</b>	<b>49</b>
<b>A</b>	<b>Tabulky</b>	<b>50</b>
<b>B</b>	<b>Grafy s výsledky rekonstrukce</b>	<b>53</b>
<b>C</b>	<b>Obsah elektronické přílohy</b>	<b>56</b>



## SEZNAM OBRÁZKŮ

3.1	DCT (64×64) . . . . .	21
3.2	Přeskládaný DCT (64×64) . . . . .	21
5.1	Dekorelace dvou atomů . . . . .	29
5.2	Slovník K-SVD . . . . .	31
5.3	Inkoherentní slovník . . . . .	31
6.1	Segmentace na stacionární úseky pro hudební signál, žánr jazz . . . .	33
8.1	Rozložení děr na popové skladbě . . . . .	40
8.2	Příklad rekonstrukce skladby pop . . . . .	41
8.3	Výběr 30 % stacionárních segmentů ze skladby žánru Country . . . .	42
B.1	Výsledky rekonstrukce na malé sadě děr pro první polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD . .	53
B.2	Výsledky rekonstrukce na malé sadě děr pro druhou polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD . .	53
B.3	Výsledky rekonstrukce na velké sadě děr pro první polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD . .	54
B.4	Výsledky rekonstrukce na velké sadě děr pro druhou polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD . .	54
B.5	Srovnání první poloviny žánrů, vztah SNR pro různé úrovně maxi- mální koherence u INK-SVD . . . . .	55
B.6	Srovnání druhé poloviny žánrů, vztah SNR pro různé úrovně maxi- mální koherence u INK-SVD . . . . .	55

# SEZNAM TABULEK

4.1	Shrnutí K-SVD algoritmu . . . . .	25
5.1	INK-SVD dekorelace . . . . .	30
5.2	INK-SVD přiřazení . . . . .	30
6.1	Algoritmus pro výběr segmentů podle koeficientů . . . . .	34
8.1	Seznam děr pro experimenty . . . . .	40
A.1	Srovnání kvality rekonstrukce na celé skladbě – sada malých děr . . .	50
A.2	Srovnání kvality rekonstrukce na celé skladbě – sada velkých děr . . .	50
A.3	Srovnání rekonstrukce pro nejlepší parametry trénovacích algoritmů– sada malých děr . . . . .	51
A.4	Srovnání rekonstrukcí pro různé slovníky s metodou PEMO-Q na sadě velkých děr . . . . .	52

# ÚVOD

Řídké reprezentace jsou v současné době využívány v různých oblastech vědy. Jednou z nich je práce se zvukovými či obrazovými signály. Typické aplikace jsou pak odstraňování šumu, komprimované snímání a spousta dalších. Na řídkých reprezentacích je založen také algoritmus OMP (Orthogonal Matching Pursuit) využitý k dopňování shluků chybějících vzorků v signálu. Rekonstrukce poškozených nebo chybějících oblastí ve zvukovém signálu se nazývá Audio Inpainting.

Nedílnou součástí algoritmu OMP je slovník. Jedná se o matici, jejíž sloupce představují frekvence. Jedním z nejpoužívanějších je DCT. Existují ale i komplexnější řešení a tím je například algoritmus K-SVD. Tato metoda se po předložení sady trénovacích dat postará o optimalizaci slovníkové matice na požadovaný signál. Matice se tedy definovaná data naučí. Při zavedení dodatečných pravidel lze K-SVD využít pro trénování inkoherentních slovníků.

Učení slovníků vyžaduje vstupní data. Jedná se o matici s dílčími signály trénovací množiny uložené ve sloupcích. Vhodně vybraná data pak určují kvalitu a použitelnost naučeného slovníku ke konkrétním rekonstrukcím. Trénovací signály lze sbírat z celé známé skladby případně využít metodu pro segmentování signálu a vybrat pouze vhodné úseky.

Snahou diplomové práce je praktická aplikace OMP k Audio Inpaintingu. K tomuto účelu jsou využity slovníky DCT, K-SVD a zvolený algoritmus INK-SVD (Inkoherentní K-SVD). V případě INK-SVD tato diplomová práce navazuje na výzkum disertační práce [3] z roku 2013 a aplikuje pozorované jevy a specifika na praktických experimentech.

Z jiné disertační práce [16] je zvolen algoritmus „greedy selection“ pro adaptivní segmentaci a je následně využit pro výběr stacionárních segmentů do trénovacích dat. V praktické části je výběr stacionárních segmentů testován v porovnání s výběrem z celé známé skladby. K účelu testování jsou sestaveny skripty v prostředí Matlab, které umožňují hromadné testování na žánrově odlišných hudebních skladbách.

# 1 ŘÍDKÉ REPREZENTACE SIGNÁLU

V posledních letech se do různých oborů vědy a matematiky rozšířil pojem řídké reprezentace (Sparse Representation – SR). Tento pojem úzce souvisí s nedourčeným systémem lineárních rovnic. Jedná se o soustavu, ve které je více neznámých než je v ní rovnic. Protože má soustava nekonečný počet řešení, metody založené na řídkých reprezentacích budou hledat právě ta, která mají současně nejvíce nulových koeficientů, respektive nejméně nenulových koeficientů. Taková řešení mají výhody jako větší datová komprese a snažší numerická manipulace [10]. Mimo jiné oblasti použití, jako rekonstrukce obrazových dat a odšumování signálů jsou algoritmy naznačené v práci vhodné k rekonstrukci zvukových signálů.

## 1.1 Definice základních pojmů

Problematika SR se opírá o pojmy popisované v [10]. Prvním z nich je „řídký vektor“  $\mathbf{x}$ . Ten je vyjádřen jako  $\mathbf{x} = [x_1, \dots, x_n]$  o velikosti  $N$ . Řídký vektor lze vyjádřit v lineární kombinaci se slovníkem  $\mathbf{D}$ , kde  $\mathbf{y}$  je pozorovaný signál. Tedy  $\mathbf{y}$  je vyjádřen jako soustava lineárních rovnic  $\mathbf{y} = \mathbf{D}\mathbf{x}$ , ve které  $\mathbf{y} \in \mathbb{C}^m$  a matice  $\mathbf{D} \in \mathbb{C}^{m \times N}$ .

Mezi další, v práci používané výrazy patří nosič vektoru

$$\text{supp}(\mathbf{x}) = \Omega = \{i | x_i \neq 0\}. \quad (1.1)$$

Jedná se o množinu indexů, ve kterých má vektor  $\mathbf{x}$  nenulovou hodnotu. Definuje tedy míru řídkosti vektoru  $\mathbf{x}$ . Například jako  $\text{supp}(\mathbf{x}) = \{1, 4\}$  pro vektor  $\mathbf{x} = [x_1, \dots, x_5] = [2, 0, 0, 7, 0]$ .

Pro potřeby algoritmů založených na řídkých reprezentacích se využívá určování vzdáleností vektorů. U každého vektoru je tedy možné určit jednotlivé normy. Norma je funkce, která k vektoru přiřadí hodnotu podle definice  $l_p$ . Normu vektoru lze definovat jako

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left( \sum_{i=1}^N |x_i|^p \right)^{1/p} && \text{pro } 1 \leq p \leq \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^N |x_i|^p && \text{pro } 1 < p < \infty, \\ \|\mathbf{x}\|_\infty &:= \max |x_i|, && (1.2) \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|, \\ \|\mathbf{X}\|_F &= \sqrt{\sum_{i=1}^m \sum_{j=1}^n |x_{i,j}|^2}. \end{aligned}$$

V případě algoritmů souvisejících s řídkými reprezentacemi budou nejpoužívanější normy  $l_0$ ,  $l_1$  a Frobeniova norma [8] pro matice  $\mathbf{X}$ . Tedy počet nenulových prvků vektoru  $\mathbf{x}$  pro  $l_0$  a součet délek prvků pro  $l_1$ . Jako příklad budou pro vektor  $\mathbf{x} = [0, 5, 3, -2, 0, 1, 0]$  v normě  $\|\mathbf{x}\|_0 = 4$  a v normě  $\|\mathbf{x}\|_1 = 0 + 5 + 3 + 2 + 0 + 1 + 0 = 11$ .

Dále je nutné určit stupeň řídkosti. Za  $K$ -řídký vektor se dá považovat takový vektor  $\mathbf{x} \in \mathbb{C}^N$ , který má počet nenulových složek menší nebo rovný počtu  $K$ . Tedy pokud splní podmínku

$$\|\mathbf{x}\|_0 \leq K, \quad (1.3)$$

kde  $K$  je maximální počet nenulových složek (Sparsity level). Tato hodnota se většinou používá v dále uvedených algoritmech jako pravidlo pro zastavení – tzv. „stopping rule“. Znamená to, že při dosažení stanovené řídkosti končí iterace, případně celý algoritmus.

Relativní řídkostí se rozumí poměr  $\frac{K}{N}$ . Množina všech  $K$ -řídkých vektorů se značí jako

$$\sum_K^N := \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_0 \leq K\}. \quad (1.4)$$

Chyba nejlepší aproximace  $\mathbf{x} \in \mathbb{C}^N$  v normě  $l_p$  je určena jako

$$\sigma_k(\mathbf{x}) := \sigma_k^N(\mathbf{x})_p := \inf_{\mathbf{z} \in \sum_k^N} \|\mathbf{x} - \mathbf{z}\|_p. \quad (1.5)$$

Jedná se o chybu mezi vektorem  $\mathbf{x}$  a  $K$ -řídkým vektorem. V případě, že  $\mathbf{x}$  bude přímo  $K$ -řídký vektor, bude chyba  $\sigma_k(\mathbf{x})_p$  nulová pro jakékoliv  $p$ , tedy pro libovolnou normu.

## 1.2 Řídká řešení a podmínky

Hlavní myšlenka řešení rovnice  $\mathbf{y} = \mathbf{D}\mathbf{x}$  vychází z potřeby získat co nejřidší vektor  $\mathbf{x}$ . Právě díky němu bude možné reprezentovat  $\mathbf{y}$  malým počtem rovnic. Požadované (přípustné) řešení je tedy vektor  $\mathbf{x}$  o dostatečné řídkosti ( $K$ -řídký). Základní podmínka označovaná jako  $(P_0)$  je [4][10]

$$\min_x \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \quad (1.6)$$

kde  $\mathbf{y} \in \mathbb{C}^m$  je vektor známých hodnot (pochází z obecného měřeného signálu) a  $\mathbf{D} \in \mathbb{C}^{m \times N}$  je tzv. slovník (dictionary). Sloupce slovníku se zde označují jako atomy. Možnosti použitého slovníku jsou široké a dále se jim věnuje kapitola 3. Možných

reprezentací  $\mathbf{y}$  je nekonečně mnoho ale atraktivní a numericky výhodné jsou právě ty nejjednodušší nebo alespoň ty s drobnou odchylkou  $\delta$

$$\min_x \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_p \leq \delta. \quad (1.7)$$

Problém nastává v užití nějakého z algoritmů konvexní optimalizace [10]. Protože norma  $l_0$  není konvexní funkce, nelze na ní žádný tento algoritmus aplikovat. Ovšem nejbližší norma  $l_1$  už konvexní je,

$$\min_x \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_p \leq \delta, \quad (1.8)$$

proto se hledají podmínky, při kterých se řešení obou úloh v normách shodují. Tyto podmínky jsou naznačeny v [10].

## 2 REKONSTRUKCE ZVUKOVÉHO SIGNÁLU

Tato kapitola se zabývá rekonstrukcí audio signálu, jinak označovaná jako „Audio Inpainting“. Ve zvukových vlnách se mohou objevit různé na poslech nepříjemné artefakty. Například při přenosu zvukových dat ze starých a poškozených nosičů (magnetické pásky, gramofonové desky) mohou být některé úseky signálu špatně digitalizovány a vzniknou zkreslené nebo úplně vynechané vzorky. Podobné chyby vznikají také při přenosu signálu přenosovým médiem (například při ztrátě paketů u internetové VoIP telefonie).

Chyby v audio signálu se projeví různými způsoby. Mezi nejčastější patří šum s praskáním, přebuzené vlny a u VoIP telefonie výpadky. V takových případech lze zaznamenaný signál vyhodnotit, tyto artefakty najít a následně rekonstruovat. Rekonstrukcí se myslí vhodnou technikou signál opravit nebo docílit co největší podobnosti s původním nepoškozeným signálem. Snaha o docílení nejbližší podobnosti nemusí být ale tou největší prioritou. Důležité je aby byl výsledek „poslouchatelný“. Tedy aby rekonstruované oblasti nepůsobili rušivým dojmem a posluchač tato místa nerozeznal.

Techniky rekonstrukce řešené v práci jsou založeny na řídkých reprezentacích signálu. Tedy nastíněné algoritmy se snaží o co nejvhodnější aproximace slovníkových atomů řídkým vektorem a tím dosažení rekonstrukce poškozené části zvukového signálu.

### 2.1 Výběr poškozených vzorků

Uvažovaný audio signál je ovlivněn pouze v některých místech. Aby se dala opravit, musí se tyto části detekovat. Problematikou detekce poškozených úseků se ale práce nezabývá. Masku může definovat přímo uživatel, případně ji sestaví externí program.

Poškozený signál definovaný  $\mathbf{y} \in \mathbb{R}^L$  vychází z původního neovlivněného vektoru  $\mathbf{s} \in \mathbb{R}^L$ . Po navzorkování průběhu je  $I \in \mathbb{N}^L$  definovaná jako množina indexů vzorků. Poté  $I^m$  jsou poškozené vzorky,  $I^r$  nepoškozené vzorky, přičemž  $\{I^r, I^m\} \in I$ . Indexy následně určují nepoškozený vzorek podle

$$\mathbf{y}^r = \mathbf{y}(I^r) = \mathbf{s}(I^r). \quad (2.1)$$

Naproti tomu poškozený vzorek bude definován jako

$$\mathbf{y}^m = \mathbf{y}(I^m) = \mathbf{s}(I^m) + \mathbf{n}.$$

Výběr ze skupiny vzorků se dá jednoduše uskutečnit pomocí  $\mathbf{M}^r$  a  $\mathbf{M}^m$

$$\begin{aligned} \mathbf{y}^r &= \mathbf{M}^r \mathbf{y} = \mathbf{M}^r \mathbf{s}, \\ \mathbf{y}^m &= \mathbf{M}^m \mathbf{y} = \mathbf{M}^m \mathbf{s} + \mathbf{n}. \end{aligned} \quad (2.2)$$

Matice zde vlastně plní funkci masky. Pro výběr spolehlivých vzorků bude celý vektor vynásoben  $\mathbf{M}^r$ . Sestaví se jako jednotková diagonální matice o rozměru  $L \times L$  s hodnotou jedna v diagonále na řádce  $\mathbf{M}^r$ , pokud  $I_i = I_i^r$ . Pro  $\mathbf{M}^m$  platí obdobně, když  $I_i = I_i^m$ .

Tedy jako příklad vytvoření takové matice  $\mathbf{M}$  je k dispozici vektor  $\mathbf{y}$  o velikosti  $L = 6$ , který vychází z poškozeného signálu. Jeho průběh je po analyzování  $\mathbf{y} = [I_1^r, I_2^m, I_3^r, I_4^r, I_5^m, I_6^m]$ . Pro nezasažené vzorky lze sestavit  $I^r = [1, 0, 1, 1, 0, 0]$  a z ovlivněných  $I^m = [0, 1, 0, 0, 1, 1]$ . Na základě podobnosti, z jednotkové matice o rozměru  $6 \times 6$  tedy  $\mathbf{I}_6$  a vzorků  $I^r$   $I^m$ , jsou sestaveny matice

$$\mathbf{M}^r = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.3)$$

$$\mathbf{M}^m = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.4)$$

## 2.2 Aproximační algoritmy

Způsobů zjišťování řídkých vektorů je celá řada. Vyjádření  $\hat{\mathbf{x}}$  je

$$\hat{\mathbf{x}} = \arg_{\mathbf{x}} \min \|\mathbf{x}\|_0 \text{ na základě } \|\mathbf{y}^r - \mathbf{M}^r \mathbf{D} \mathbf{x}\|_2^2 \leq \theta^\epsilon. \quad (2.5)$$

Problém je ale v řešení tohoto vektoru. Přestože je vztah přímo neřešitelný, může být nahrazen aproximativním algoritmem. Známých a použitelných metod je v současné době více jak je uvedeno například v [10]. V podstatě by se daly rozdělit do dvou skupin – algoritmy vycházející z  $l_1$ -relaxace a tzv. „hladové“ (Greedy) algoritmy.

Mezi relaxační patří například:

- BP (Basis Pursuit),
- modifikovaný LARS (Least Angle Regression homothropy method),
- IRLS (Iterative Reweighted Least Squares),
- Dantzing Selector.



Jejich princip spočívá v hledání relativně blízkého řešení za podmínek  $l_1$ -relaxace, jak je naznačeno v kapitole 1.2.

Hladové algoritmy jsou realizovány především různými modifikacemi Matching Pursuit [4][7]:

- OMP (Orthogonal Matching Pursuit),
- MP (Matching Pursuit),
- Weak-MP (Weak Matching Pursuit),
- LS-OMP (Least Squares Orthogonal Matching Pursuit).

Mezi hlavní výhodu hladových algoritmů patří nízká složitost. Největší nevýhoda je zase neschopnost zajištění globálního optima. Z vypsanych hladových algoritmů jsou všechny založeny na podobném principu popsaném v kapitole 2.2.1.

### 2.2.1 Algoritmus OMP

Metoda pro výpočet řídkého vektoru  $\mathbf{x}_i$ , OMP je založena na aproximaci, kdy se při každé iteraci vybírá jeden (případně více) atomů. Tyto atomy se při dalším průchodu podílejí na výsledném řešení.

Úkolem algoritmu je tedy najít řešení  $(P_0) : \min_x \|\mathbf{x}\|_0$  vzhledem k  $\mathbf{y}^r = \mathbf{M}^r \mathbf{D} \mathbf{x}$  [1] [7]. Nejprve jsou programu dodány vstupní parametry: vektor nepoškozeného průběhu,  $\mathbf{M}^r$  pro vektor  $\mathbf{y}^r$ , maximální počet nenulových složek  $\mathbf{x}$  viz (1.3) a chyba nejlepší aproximace viz (1.5).

Prvním krokem je určení počátečních podmínek. Slovník  $\tilde{\mathbf{D}}$  vznikne vynásobením  $\mathbf{M}^r$ , původní podobou slovníku  $\mathbf{D}$  a maticí  $\mathbf{W}$ . Diagonální matice  $\mathbf{W}$  je složena z prvků  $\mathbf{W}_{jj} = \|\mathbf{M}^r \mathbf{d}_j\|_2^{-1}$  v hlavní diagonále. Ostatní buňky  $j \neq j'$  jsou rovny nule  $\mathbf{W}_{jj'} = 0$ . Proměnná  $k$  je zde použita jako čítač iterace.

Konec podmínkového cyklu nastane v případě splnění jedné ze dvou podmínek. První bude splněna za předpokladu, že cyklus proběhne tolikrát, aby algoritmus našel vektor  $\mathbf{x}_i$  s řídkostí menší než  $K_{\max}$ . Druhá podmínka zastaví cyklus pokud bude reziduum v  $l_2$  normě menší, než definovaná chyba nejlepší aproximace  $\theta^\epsilon$ . Jinými slovy bude buď dosaženo vektoru o dostatečné řídkosti nebo chyby o přijatelné velikosti [5].

## 2.3 Hodnocení kvality rekonstrukce

K posouzení kvality rekonstrukce jsou v následujících odstavcích představeny dvě metody. Obě z nich jsou objektivní metody hodnocení. První SNR měří podobnost mezi dvěma signály. Druhá metoda PEMO-Q je komplexnější a hodnocené signály procházejí navíc sadou filtrů, které simulují lidský sluchový aparát.

### 2.3.1 Výpočet SNR

Ke srovnání kvality rekonstrukce může být použito SNR (odstup signálu od šumu – Signal-to-Noise Ratio). V případě hodnocení kvality rekonstrukce SNR reprezentuje spíše „odstup rekonstruovaného od originálního signálu“ [1]. Vypočítá se z opraveného signálu reprezentovaného vektorem  $\hat{\mathbf{y}}(I)$  a původního  $\mathbf{y}(I)$  jako

$$\text{SNR}_e(\mathbf{y}, \hat{\mathbf{y}}) = 10 \log \frac{\|\mathbf{y}(I)\|_2^2}{\|\mathbf{y}(I) - \hat{\mathbf{y}}(I)\|_2^2}. \quad (2.6)$$

Výsledek je číslo v jednotkách decibelů. Čím větší bude tím shodnější oba signály jsou. Pokud budou oba totožné, výsledek se bude blížit nekonečnu.

Problém SNR je, že metoda do výsledku uvažuje pouze odstup dvou signálů. Tedy signály neprocházejí žádnými filtry a nevyužívá se žádný akustický model. Nelze tedy jednoznačně tvrdit, že rekonstrukce s lepší hodnotou SNR bude na poslech méně rušivá.

### 2.3.2 PEMO-Q

Další metodou ke srovnání a vyhodnocení kvality rekonstrukce je PEMO-Q. Oproti SNR se PEMO-Q snaží do výsledku zahrnout psychoakustický model. Vychází se z referenčního a poškozeného signálu a oba jsou dále zpracovány na základě lidského sluchového modelu. Výstupy jsou po průchodu sadou filtrů křížově korelovány a případně i váhovány a kvantovány. Samotné zpracování signálů je poměrně komplexní a do detailu jej popisuje [9].

Výstupem jsou dvě hodnoty. První Perceptual Similarity Measure (PSM) a druhá Temporary Perceptual Similarity Measure (PSMt) jsou čísla, která na stupnici představují ohodnocení kvality rekonstrukce. Narozdíl od PSM jsou data u PSMt zpracovávány v 10 ms segmentech a v poslední části zpracování navíc dochází k váhování a kvantování. Díky tomu má PSMt větší nezávislost na typu a charakteru signálu, zatímco výsledek PSM je jím výrazněji ovlivněn. Obě čísla vycházejí v rozmezí  $[-1, 1]$ . Pokud se výsledek blíží  $-1$ , znamená to špatnou rekonstrukci a naopak hodnoty blízké  $1$  představují perfektní rekonstrukci.

## 3 SLOVNÍKY

Pro aproximaci signálu algoritmy dále uvedenými v práci se využívá slovník  $\mathbf{D}$ . Slovník je matice hodnot (diskrétních frekvencí). Jednotlivé sloupce tvořící slovník se nazývají atomy. Volba vhodného slovníku je klíčová při snaze najít co možná nejjednodušší  $\mathbf{x}$  pro reprezentaci libovolného signálu podle  $\mathbf{y} = \mathbf{D}\mathbf{x}$ . Způsoby k jejich získání lze rozdělit do dvou základních skupin.

- Statické slovníky (DCT)
- Naučené slovníky (algoritmy K-SVD, MOD, Inkoherentní K-SVD, RLS-DLA)

Oba typy mají své výhody a nevýhody. V diplomové práci jsou k aplikaci na rekonstrukce vybrány DCT, K-SVD a inkoherentní K-SVD.

### 3.1 Statické slovníky

Statický slovník je matice diskretních koeficientů, které jsou sestaveny z pevné funkce. Tato transformační funkce v podstatě určuje vlastnosti samotného slovníku. Jednotlivé atomy v sobě nesou potřebné informace, jako jsou například frekvence a fáze. S větším množstvím těchto informací teoreticky rostou i možnosti a přesnost rekonstrukce libovolného signálu. Matice se v některých zdrojích označuje také jako jádro transformace. Jejich výhodou je rychlé sestavení matice a univerzálnost pro odlišné signály. Zároveň může sloužit jako inicializační slovník při adaptaci na specifický signál.

#### 3.1.1 DCT slovník

Typický a často používaný zástupce statického slovníku je DCT (Discrete Cosine Transform). Vyniká především svou jednoduchostí a dobrou sestavení. Jeho matice je složena ze sloupcových vektorů podle funkce  $\mathbf{D} = [\mathbf{d}_0, \dots, \mathbf{d}_{K-1}]$ . Tyto sloupce  $j$  jsou z rozsahu  $0 \leq j \leq K - 1$ . Pro každý sloupec o délce  $N$  platí  $0 \leq t \leq N - 1$  jako

$$\mathbf{d}_j(t) = \mathbf{w}_d(t) \cos\left(\frac{\pi}{K} \left(t + \frac{1}{2}\right) \left(j + \frac{1}{2}\right)\right), \quad (3.1)$$

kde parametr  $K$  definuje počet sloupců matice. Prakticky tedy slovník udává  $K$  frekvencí. Násobící okno  $\mathbf{w}_d$  může být realizováno například sinusovým násobícím okénkem.

Slovník DCT ve své definici nese tedy pouze informace o frekvencích. Díky tomu je často používán v oblasti komprese zvukových a obrazových signálů. Jeho aplikací je dosaženo kompaktní spektrální reprezentace [11][1]. Uvedená definice je přesněji DCT-IV. V praktické části se používá ortogonální verze DCT, tedy slovníková matice je rozměrem symetrická.

## 3.2 Naučené slovníky

Naučené slovníky (nebo také adaptované slovníky) jsou vytvořené na míru konkrétnímu signálu nebo signálům. Prakticky se používají algoritmy, které použijí inicializační slovník jako matici a následně ji natrénují sadou trénovacích dat. Výstupem je specifický slovník optimalizovaný pro konkrétní signál. Další možností je sestavení matice ze sady statických funkcí a podle charakteru signálu (například rozdělení segmentu do více samostatných částí) jsou do atomů vybrány ty nejvhodnější. Oba tyto druhy mají již široké využití v kódování obrazových dat a diplomová práce se zaměří na užití trénovacích algoritmů pro zvukové signály.

### 3.2.1 K-SVD

Metoda K-SVD pro přizpůsobení slovníků vychází z principu zobecněného algoritmu K-means pro kvantizaci vektorů a SVD (singular value decomposition – singulární rozklad) pro rozklad signálu. Úprava atomů probíhá ve dvou fázích. Nejprve se zafixuje počáteční vygenerovaná matice slovníku a hledá se řídký vektor  $\mathbf{x}$  při dané sadě vzorků signálu, následně se určí neměnné  $\mathbf{x}$  a hledá se vhodná kombinace atomů [2]. Detailnímu popisu algoritmu se věnuje kapitola 4.

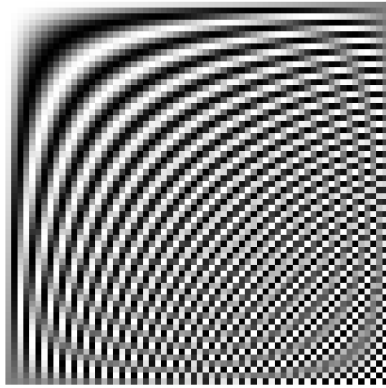
### 3.2.2 Inkoherentní K-SVD

Algoritmus K-SVD neřeší, jak moc jsou si atomy uvnitř matice podobné. Může tedy nastat situace, kdy pro dva různé atomy bude vysoká vzájemná koherence. Ta určuje právě míru podobnosti mezi dvěma atomy. Z hlediska použití rekonstrukce na bázi řídkých algoritmů je to ale nevýhodné. Protože je k rekonstrukci vybíráno co nejméně možných atomů, je zde požadavek na jejich odlišnost. Na této myšlence stojí inkoherentní algoritmy pro učení slovníku. V něm se technika K-SVD rozšíří o dekořelační algoritmus, který provede vzájemné odlišení atomů [14]. Inkoherentním algoritmům se podrobněji věnuje kapitola 5.

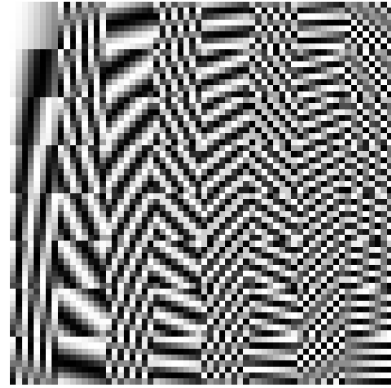
## 3.3 Vizualizace slovníků

Slovníkové matice představené v práci mají své atomy uložené v jednotlivých sloupcích. Tyto hodnoty se dají poměně jednoduše zobrazit v obrazových maticích. Nejprve jsou hodnoty normovány v rozsahu 0–225, kde 0 představuje černou a číslo 255 bílou barvu.

Na obrázku 3.1 je potom takto zobrazený DCT slovník o velikosti  $64 \times 64$ . Jsou v něm dobře vidět jednotlivé báze. Horší je to se slovníky, které nemají pravidelnou



Obr. 3.1: DCT ( $64 \times 64$ )



Obr. 3.2: Přeskládaný DCT ( $64 \times 64$ )

strukturu jako je například výsledek po K-SVD. V takovém případě jsou atomy, rozložené v celém sloupci hůře čitelné a porovnatelné. Proto se používá přeskládání do submaticových oblastí (takzvané „patche“). Příklad DCT přeskládaného do oblastí je na obrázku 3.2. Zde se nachází stejný slovník, jehož atomy jsou sestavené do čtverců o velikosti  $8 \times 8$  (jeden atom je v jednom čtverci). I zde se dají báze poměrně dobře rozeznat. Díky této podobě lze ale i lépe lokalizovat odlišnosti uvnitř čtverce a případné artefakty u matic s méně pravidelnou strukturou. Uvedený způsob skládání atomů a následné zobrazení se často používá pro slovníky určené k práci s obrazem. Dá se ale i dobře využít při užití pro zvukové signály.

## 4 ALGORITMUS K-SVD

Představa využití K-SVD vychází z optimalizace libovolného počátečního slovníku a to podle definovaných případně vypořizovaných parametrů. Jde tedy o extrakci parametrů z cílového signálu a jejich následné použití jako vstupní parametry pro učení.

### 4.1 Kvantizace vektorů a K-means

K-SVD algoritmus se zakládá na problematice kvantizace vektoru VQ (vector quantization nebo také „clustering“). Kvantizace vektorů zde představuje extrémní případ řídké reprezentace. V rozkladu signálu se sada deskriptivních vektorů  $\{\mathbf{d}_k\}_{k=1}^K$  natrénuje na signál. Každý vzorek signálu je tedy reprezentován pouze jedním z těchto vektorů.

Trénování K-SVD vychází z principu zobecněného „K-means“. Algoritmu se také říká Generalized Lloyd Algorithm (GLA). V případě K-SVD se zobecněním myslí použití více vektorů k vyjádření signálu (ale pořád řídkých).

Účel je ve dvou krocích najít atomy  $\mathbf{D}$ , vhodné k reprezentaci signálu. V prvním kroku je snaha o aproximaci sady vzorků  $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$  ( $N \gg K$ ) hledáním řídkého vektoru. Počet atomů  $\mathbf{D}$  je  $K$ , počet vzorků signálu  $N$ . Druhý krok hledá aktualizace atomů pro již získané reprezentační vektory  $\mathbf{x}_i$ .

### 4.2 Popis algoritmu

Měření kvality reprezentace signálu se určuje podle matice reziduí  $\mathbf{E}$ , kde

$$\mathbf{E} = \sum_{i=1}^K e_i^2 = \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2. \quad (4.1)$$

Rezidua zde představují číselnou chybu reprezentace signálu pomocí slovníku  $\mathbf{D}$  a matice řídkých vektorů  $\mathbf{X}$ . S lépe reprezentujícím  $\mathbf{D}$  se chyby v matici zmenšují. Úkolem K-SVD je zajištění co nejmenší chyby mezi sadou vzorků  $\mathbf{Y}$  a lineární kombinací  $\mathbf{D}$  s  $\mathbf{X}$  [2].

#### 4.2.1 Použití K-means

Hlavní podmínka pro optimalizaci slovníku je tedy

$$\min_{\mathbf{D}, \mathbf{X}} \{\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2\} \text{ na základě } \forall i, \|\mathbf{x}_i\|_0 \leq T_0. \quad (4.2)$$

Parametr  $T_0 = |\text{supp}(\mathbf{x})|$  představuje míru řídkosti  $\mathbf{x}_i$ . Protože je nalezení  $\mathbf{D}$  a  $\mathbf{X}$  komplexní problém, hledání se rozdělí do dvou kroků.

1. Pevný  $\mathbf{D}$  a hledá se  $\mathbf{X}$ .
2. Pevné  $\mathbf{X}$  a hledá se  $\mathbf{D}$ .

V první fázi jde o nalezení  $\mathbf{X}$  složené z řídkých vektorů o řídkosti  $T_0$ . Problém lze vyjádřit jako funkci

$$\min_{\mathbf{x}_i} \{ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 \} \text{ na základě } \|\mathbf{x}_i\|_0 \leq T_0 \quad (4.3)$$

pro  $i = 1, 2, \dots, N$ .

Řešení je s pomocí OMP jednoduché. Stačí použít aproximační algoritmus k nalezení  $\mathbf{x}$  a s požadavkem na dostatečnou řídkost jsou vektory získány.

Druhá fáze je ale o něco složitější. Zde je požadavkem vhodně upravit atomy  $\mathbf{D}$ . K dosažení co nejmenší chyby – reziduí 4.1 je nutné rozložit kombinaci  $\mathbf{DX}$  podle

$$\begin{aligned} \|\mathbf{Y} - \mathbf{DX}\|_F^2 &= \left\| \mathbf{Y} - \sum_{j=1}^K \mathbf{d}_j \mathbf{x}_T^j \right\|_F^2 \\ &= \left\| \left( \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j \right) - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \\ &= \left\| \mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2. \end{aligned} \quad (4.4)$$

Vektor  $\mathbf{x}_T^k$  představuje  $k$ -tou řadu matice  $\mathbf{X}$  a vektor  $\mathbf{d}_k$  atom slovníku jemu odpovídající. Jak je naznačeno v 4.4 došlo k rozložení násobku  $\mathbf{DX}$  do součtu  $K$  vektorů. Výsledkem je chybová matice  $\mathbf{E}_k$  pro všechny  $N$  vzorky ale bez  $k$ -tého atomu.

Následně se definuje  $\boldsymbol{\omega}_k$  jako vektor indexů  $i$  ukazující na nenulové prvky  $\mathbf{x}_T^k$ .

$$\boldsymbol{\omega}_k = \{i | 1 \leq i \leq K, \mathbf{x}_T^k(i) \neq 0\}. \quad (4.5)$$

Z těchto ukazatelů se sestaví matice  $\boldsymbol{\Omega}_k$  o velikosti  $N \times |\boldsymbol{\omega}_k|$  s jedničkami na pozicích  $\Omega(\omega_k(i), i)$  a nulami všude jinde. Vznikne vlastně maska na nenulové prvky a po vynásobení s  $\mathbf{x}_R^k = \mathbf{x}_T^k \boldsymbol{\Omega}_k$  bude získán vektor  $\mathbf{x}_R^k$  o délce  $|\boldsymbol{\omega}_k|$ . Obdobně pro  $\mathbf{Y}_k^R = \mathbf{Y} \boldsymbol{\Omega}_k$  vznikne „podsada“ vzorků aktuálně používané atomem  $\mathbf{d}_k$  s rozměrem  $n \times |\boldsymbol{\omega}_k|$ . Stejně s chybovou maticí  $\mathbf{E}_k^R = \mathbf{E}_k \boldsymbol{\Omega}_k$ .

Tímto postupem dojde k výběru atomů, které se budou účastnit aktualizace. Především ale byly odstraněny nulové a nezajímavé prvky z  $\mathbf{x}_T^k$ . Při následném rozkladu matice  $\mathbf{E}_k^R$  pomocí SVD bude mít řešené  $\tilde{\mathbf{x}}_T^k$  zajištěnou stejnou řídkost jako  $\mathbf{x}_T^k$  [2].

## 4.2.2 Použití SVD

Nyní se algoritmus zaměří na vhodnou úpravu atomů  $\mathbf{d}_k$ . Momentálně je dosaženo stavu, kdy lze minimalizaci vyjádřit jako

$$\|\mathbf{E}_k \boldsymbol{\Omega}_k - \mathbf{d}_k \mathbf{x}_T^k \boldsymbol{\Omega}_k\|_F^2 = \|\mathbf{E}_k^R - \mathbf{d}_k \mathbf{x}_R^k\|_F^2. \quad (4.6)$$

Jak aktualizovat atomy  $\mathbf{d}_k$  ze slovníku  $\mathbf{D}$  se zjistí z rozložení chybové matice  $\mathbf{E}_k^R$ .

Rozložení Singulární rozklad (SVD). Po aplikaci dojde k rozebrání  $\mathbf{E}_k^R$  do tří matic

$$\mathbf{E}_k^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \in \mathbb{R}^{n \times |\omega_k|}, \quad (4.7)$$

kde  $\mathbf{U} \in \mathbb{R}^{n \times n}$  a  $\mathbf{V} \in \mathbb{R}^{|\omega_k| \times |\omega_k|}$  jsou unitární matice a  $\mathbf{\Sigma}$  je diagonální matice s kladnými reálnými prvky. Podrobný popis singulárního rozkladu matic lze nalézt v [8].

Získat řešení je nyní jednoduché. Aktualizace pro  $k$ -tý atom je první sloupec z  $\mathbf{U}$ . Vektor  $\mathbf{x}_R^k$  je násobek prvního sloupce  $\mathbf{V}$  a  $\mathbf{\Sigma}(1, 1)$

$$\tilde{\mathbf{d}}_k = \mathbf{U}(k, 1) \quad (4.8)$$

$$\mathbf{x}_R^k = \mathbf{V}(k, 1)\mathbf{\Sigma}(1, 1). \quad (4.9)$$

Dále je nutné podotknout, že algoritmus K-SVD je iterační. Tedy pro každou iteraci dochází k aktualizaci atomů [2]. Ke shrnutí je uveden celý algoritmus v tabulce 4.1.



Tab. 4.1: Shrnutí K-SVD algoritmu

<p><b>Vstupní parametry:</b> vzorky <math>\{\mathbf{y}_i\}_{i=1}^N</math>, počet iterací <math>J_0</math>, řídkost <math>T_0</math>,</p>
<p><b>Úkol:</b> nalezení slovníku pro nejlepší reprezentaci sady vzorků <math>\{\mathbf{y}_i\}_{i=1}^N</math> vyřešením <math>\min_{\mathbf{D}, \mathbf{X}} \{\ \mathbf{y}_i - \mathbf{D}\mathbf{x}_i\ _2^2\}</math> na základě <math>\forall i, \ \mathbf{x}_i\ _0 \leq T_0</math>.</p>
<p><b>Inicializace:</b> sestavení počátečního <math>\mathbf{D}^{(0)} \in \mathbb{R}^{n \times K}</math> a Počítadlo iterací <math>J = 0</math></p>
<p><b>Hlavní cyklus:</b> dokud není splněno <math>J = J_0</math>.</p> <p><i>První krok:</i> za pomoci OMP najít vektor <math>\mathbf{x}_i</math> ke každému vzorku <math>\mathbf{y}_i</math> pro <math>i = 1, 2, \dots, N</math>, <math>\min_{\mathbf{x}_i} \{\ \mathbf{y}_i - \mathbf{D}\mathbf{x}_i\ _2^2\}</math> na základě <math>\ \mathbf{x}_i\ _0 \leq T_0</math>.</p> <p><i>Druhý krok:</i> aktualizace každého atomu <math>k = 1, 2, \dots, K</math> slovníku <math>\mathbf{D}^{J-1}</math>. Určení skupiny vzorků, které používají aktuální atom definicí <math>\omega_k = \{i   1 \leq i \leq N, \mathbf{x}_T^k(i) \neq 0\}</math>. Zjištění chybové matice <math>\mathbf{E}_k</math> podle <math>\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j</math>. Omezení <math>\mathbf{E}_k</math> výběrem prvků určených <math>\omega_k</math> a tím získání <math>\mathbf{E}_k^R</math>. Singularní rozklad matice <math>\mathbf{E}_k^R</math> na <math>\mathbf{E}_k^R = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T</math> a z nich aktualizace <math>\tilde{\mathbf{d}}_k = \mathbf{U}(k, 1)</math>, <math>\mathbf{x}_R^k = \mathbf{V}(k, 1)\mathbf{\Sigma}(1, 1)</math>. Navýšení iteračního kroku <math>J = J + 1</math>.</p>
<p><b>Výstupní parametry:</b> <math>\mathbf{D}, \mathbf{X}</math></p>

## 4.3 Optimální parametry pro K-SVD

Podmínkou pro zajištění optimálně natrénovaného K-SVD slovníku jsou vstupní parametry algoritmu. Tyto parametry jsou závislé na charakteru signálu ale je možné zvolit obecné hodnoty. Obecné parametry jsou podrobně zkoumány a věnuje se jim článek [13]. Ze sady měření byly vyzvořovány a následně vybrány následující hodnoty.

Prvním parametrem je volba a výběr trénovacích dat. Trénovacími daty se dále zabývá kapitola 6. Se sestavováním matice trénovacích dat úzce souvisí parametr posun segmentů (shift).

Dále lze specifikovat inicializační slovník, tedy matici v kombinaci s kterou se bude počítat řídký vektor při první iteraci. Volba inicializačního slovníku nemá na výslednou podobu matice téměř vliv. Může být tedy složen z náhodných čísel. Jedinou podmínkou však je, aby všechna čísla byla normována.

Podstatným parametrem pro první fázi algoritmu je hodnota maximální řídkosti vektoru  $\mathbf{x}$ . Ze série měření v [13] vyplývá, že vhodná řídkost pro uvažované signály je 3. Znamená to, že v řídkém vektoru  $\mathbf{x}$  mohou být maximálně 3 nenulové prvky.

Nakonec je potřeba specifikovat maximální počet iterací algoritmu. Pro většinu slovníků obvykle postačuje prvních 10 iterací. Je ale vhodné algoritmu povolit alespoň 80 iterací.

## 5 UČENÍ INKOHHERENTNÍHO SLOVNÍKU

Následující kapitola se věnuje upravené variantě algoritmu K-SVD a zavádí nové úvahy a postupy do procesu učení optimálního slovníku. Mezi učení inkohherentních slovníků se dá zařadit několik skupin metod. Patří mezi ně například konstrukce tzv. „Grassmannovských framů“, dále K-SVD s aplikací dekorelačního algoritmu INK-SVD případně dekorelačního algoritmu iterativní projekce a rotace. Tyto metody by měly do výsledné naučené matice přinést odlišení atomů, od kterého se očekává pozitivní ovlivnění aproximace signálu.

### 5.1 Vzájemná koherence

V textu se často zmiňuje pojem vzájemná koherence. Tato hodnota představuje stupeň podobnosti mezi různými atomy, případně skupinou atomů. Jak uvádí [3] a především [18], vzájemná koherence  $\mu(\mathbf{D})$  se vyjádří jako maximum absolutní hodnoty ze skalárního součinu pro dva různé atomy:

$$\mu(\mathbf{D}) = \max_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|. \quad (5.1)$$

Z uvedeného vztahu vyplývá, že pokud dvě báze budou ortonormální, jejich  $\mu = 0$ . Hodnota  $\mu$  je vždy z rozsahu  $\langle 0, 1 \rangle$ . Slovník tedy můžeme označit za inkohherentní, pokud vzájemná koherence všech atomů bude menší než stanovená  $\mu$ .

### 5.2 Inkohherentní K-SVD

Jedním ze způsobů učení inkohherentních slovníků je metoda „Grassmannovských framů“. Ke konstrukci se využívá algoritmus iterativních projekcí. Práce se ovšem zaměřuje na jiný způsob vytváření slovníků. Tento způsob vychází z původního K-SVD algoritmu, který je v každé iteraci doplněn o dekorelaci slovníkových atomů. Problematika řešení K-SVD je rozšířena o následující požadavek:

$$\mathbf{D}^* = \arg \min_{\mathbf{D}} C(\mathbf{D}) \text{ na základě } \mu(\mathbf{D}) \leq \mu_0, \quad (5.2)$$

kde  $C(\mathbf{D})$  je funkce, vyjadřující kvalitu aproximace slovníku a  $\mu_0$  je stanovená nejvyšší povolená vzájemná koherence, tedy práh.

Učení inkohherentního slovníku zahrnující krok dekorelace je podobně jako u K-SVD rozděleno – zde ovšem do tří podproblémů. Začíná se u inicializačního slovníku  $\mathbf{D}^0$  a v každé iteraci se provedou tři kroky [3].

1. **Fáze řídkého kódování**, pro daný pevný slovník  $\mathbf{D}^{(j)}$  se hledá  $\mathbf{X}^{(j)}$ .

2. **Aktualizace slovníku**, pro pevně danou matici aproximačních koeficientů  $\mathbf{X}^{(j)}$  se hledá  $\tilde{\mathbf{D}}^{(j+1)}$ .
3. **Dekorelace slovníku**, pro pevné  $\mathbf{X}^{(j)}$  a  $\tilde{\mathbf{D}}^{(j+1)}$  se slovník optimalizuje deko-relačním algoritmem podle vztahu 5.2 na  $\mathbf{D}^{(t+1)}$ .

V každé iteraci je tedy výsledkem slovník se vzájemnou koherencí pod stanovenou hranicí.

## 5.3 Algoritmus INK-SVD

Z metod pro učení inkoherentních slovníků se práce zaměří na INK-SVD (Inkoherentní K-SVD). Výhodou INK-SVD oproti druhému algoritmu – IPR (Iterativní projekce a rotace) je menší výpočetní náročnost. Už samotné rekonstrukce jsou časově náročné a učení inkoherentního algoritmu tuto dobu násobně prodlužuje. Z toho důvodu byl pro rekonstrukce zvolen INK-SVD.

Problém se skládá z nalezení nejbližšího slovníku  $\mathbf{D}$  k danému slovníku  $\tilde{\mathbf{D}}$  podle Frobeniovy normy, vzhledem k podmínce vzájemné koherence určené  $\mu_0$ . Vztah z rovnice 5.2 se dá přepsat jako

$$\mathbf{D}^* = \arg \min_{\mathbf{D}} \|\tilde{\mathbf{D}} - \mathbf{D}\|_F \text{ na základě } \mu(\mathbf{D}) \leq \mu_0, \quad (5.3)$$

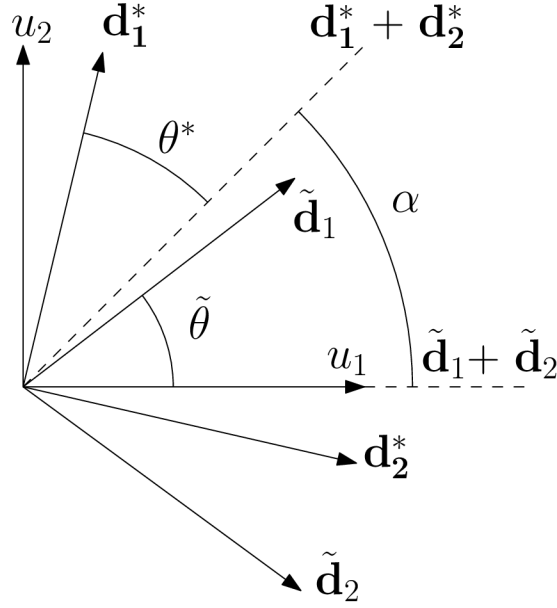
kde  $\tilde{\mathbf{D}}$  je matice, která vznikne hned po druhé fázi učení – aktualizaci slovníku [14] [3].

### 5.3.1 Dekorelace dvou atomů

Aby se dal algoritmus aplikovat na celý slovník, proces se rozdělí na dílčí části. Nejprve se zaměřit na deko-relaci dvou atomů. Tyto atomy budou označeny jako  $\tilde{\mathbf{d}}_1$  a  $\tilde{\mathbf{d}}_2$  a předpokládá se, že jejich  $\mu > \mu_0$ . Dále je uvažován  $\tilde{\theta}$  jako poloviční úhel mezi  $\tilde{\mathbf{d}}_1$  a  $\tilde{\mathbf{d}}_2$ . Obdobně pak  $\theta^*$  jako poloviční úhel mezi  $\mathbf{d}_1^*$  a  $\mathbf{d}_2^*$ . Pro lepší představu jsou jednotlivé atomy a úhly naznačeny v obrázku 5.1.

V obrázku jsou zároveň vyznačeny součty atomů  $\tilde{\mathbf{d}}_1 + \tilde{\mathbf{d}}_2$  a  $\mathbf{d}_1^* + \mathbf{d}_2^*$  a především úhel  $\alpha$ , který mezi sebou svírají. V orthonormální bázi platí, že

$$(\mathbf{u}_1, \mathbf{u}_2) = \left( \frac{\tilde{\mathbf{d}}_1 + \tilde{\mathbf{d}}_2}{\|\tilde{\mathbf{d}}_1 + \tilde{\mathbf{d}}_2\|_2}, \frac{\tilde{\mathbf{d}}_1 - \tilde{\mathbf{d}}_2}{\|\tilde{\mathbf{d}}_1 - \tilde{\mathbf{d}}_2\|_2} \right). \quad (5.4)$$



Obr. 5.1: Dekorelace dvou atomů

Uvažované vektory atomů se dají za pomoci výše zmíněných uhlů vyjádřit jako

$$\tilde{\mathbf{D}} = (\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2) = \begin{pmatrix} \cos \tilde{\theta} & \cos \tilde{\theta} \\ \sin \tilde{\theta} & -\sin \tilde{\theta} \end{pmatrix} \quad (5.5)$$

$$\mathbf{D}^* = (\mathbf{d}_1^*, \mathbf{d}_2^*) = \begin{pmatrix} \cos \alpha + \theta^* & \cos \alpha + \theta^* \\ \sin \alpha + \theta^* & -\sin \alpha + \theta^* \end{pmatrix}. \quad (5.6)$$

Podmínka pro vzájemnou koherenci z rovnice 5.3 a definice 5.1 se dá přepsat jako omezení v následujícím tvaru:

$$|\langle \mathbf{d}_1^*, \mathbf{d}_2^* \rangle| = |\cos 2\theta^*| \leq \mu_0, \quad (5.7)$$

kde se hlavní funkce vyjádří jako

$$\|\tilde{\mathbf{d}}_1, \mathbf{d}_1^*\| = 2 - 2 \cos(\tilde{\theta} - \theta^* - \alpha), \quad (5.8)$$

$$\|\tilde{\mathbf{d}}_2, \mathbf{d}_2^*\| = 2 - 2 \cos(\tilde{\theta} - \theta^* + \alpha), \quad (5.9)$$

$$\|\tilde{\mathbf{D}}, \mathbf{D}^*\| = 4 - 4 \cos(\tilde{\theta} - \theta^*) \cos(\alpha). \quad (5.10)$$

Výsledná funkce ze vztahu 5.8 bude minimální při předpokladu  $\cos(\tilde{\theta} - \theta^*)$  právě pokud  $\alpha = 0$  a zároveň úhly  $\tilde{\theta}$  a  $\theta^*$  si budou co nejvíce podobné. Problém zajištění koherence bude vyřešen rotací atomů  $\mathbf{d}_1$  a  $\mathbf{d}_2$  symetricky o úhel  $\theta^*$ . Tento úhel se zjistí jako:

$$\cos 2\theta^* = \mu_0 \quad (5.11)$$

$$\theta^* = \frac{\arccos \mu_0}{2}. \quad (5.12)$$

### 5.3.2 Dekorelace celého slovníku

Pro dekorrelaci celého slovníku je nutné zajistit nejnižší povolenou koherenci mezi všemi páry atomů ve slovníku. Pokud se budou dekorrelovat atomy postupně, tzn. vybere se jeden pár, dekorreluje a znovu, změní se zároveň hodnota koherence pro pár vybraného v kombinaci s jiným. Autor disertační práce [3] se rozhodl problém řešit a především urychlit paralelním zpracováním. Vždy se vyberou páry s největší koherencí mezi sebou, přiřadí se do množiny zpracovávaných atomů a následně se všechny dekorrelují. Průběh algoritmu je uveden v tabulkách 5.1 a 5.2.

Hlavní část algoritmu ukazuje tabulka 5.1, ve které je shrnutý již výše zmíněný postup dekorrelace atomů. Druhá tabulka 5.2 potom ukazuje výběr atomů s nejvyšší

Tab. 5.1: INK-SVD dekorrelace

<b>Vstupní parametry:</b> $\tilde{\mathbf{D}}, \mu_0$
<b>while</b> $\mu(\tilde{\mathbf{D}}) > \mu_0$ <b>do</b> $E = \text{přiřazení}(\tilde{\mathbf{D}}, \mu_0);$ <b>for</b> $\forall(\mathbf{d}_i, \mathbf{d}_j) \in E$ <b>do</b> dekorrelovat( $\mathbf{d}_i, \mathbf{d}_j$ ); <b>end</b> <b>end</b>
<b>Výstupní parametr:</b> $\mathbf{D}^*$

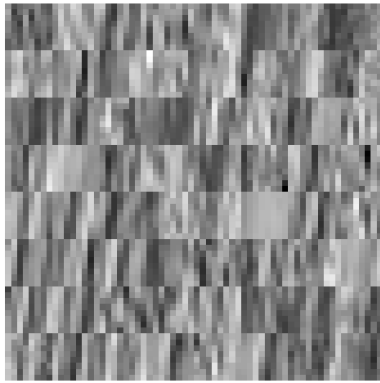
vzájemnou koherencí a jejich párové přiřazování do  $E$ . Parametr  $E$  zde představuje množinu párových atomů.

Tab. 5.2: INK-SVD přiřazení

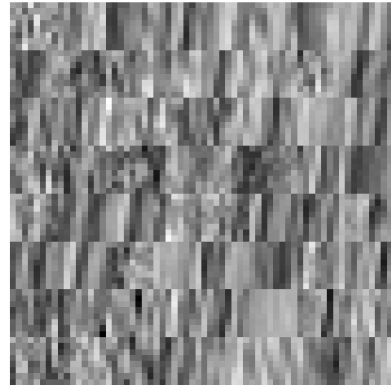
<b>Vstupní parametry:</b> $\tilde{\mathbf{D}}, \mu_0$
// Inicializace $\mathbf{D} \leftarrow \tilde{\mathbf{D}}$ $E \leftarrow \emptyset$ <b>while</b> $\mu(\mathbf{D}) > \mu_0$ <b>do</b> $(i, j) = \arg \max_{i,j}  (\mathbf{D}^T \mathbf{D} - \mathbf{I})_{i,j} ;$ $\mathbf{D} \leftarrow \mathbf{D} \setminus \{\mathbf{d}_i, \mathbf{d}_j\};$ $E \leftarrow E \cup \{(\mathbf{d}_i, \mathbf{d}_j)\}$ <b>end</b>
<b>Výstupní parametr:</b> $E$

### 5.3.3 Vizualizace odlišnosti učených slovníků

Obrázky 5.2 a 5.3 jsou příklady zobrazení K-SVD a inkoharentního slovníku. Jedná se o slovník naučený v obou případech stejným postupem, stejnými trénovacími daty a stejnými parametry, jen ve druhém případě byl aplikován dekorelační algoritmus. Při bližším zkoumání je ale vidět, že se atomy mezi oběma obrázky příliš nepodobají. Účelem dekorelačního algoritmu je zajistit vzájemnou koherenci mezi atomy pod stanovenou úroveň. To znamená, že atomy které jsou si příliš podobné budou upraveny. Na obrázku 5.3 je nastavená vzájemná koherence  $\mu_0 = 0,1$ , atomy jsou tedy výrazně upravené a proto se od slovníku naučeného s K-SVD liší.



Obr. 5.2: Slovník K-SVD



Obr. 5.3: Inkohorentní slovník

## 6 METODY SBĚRU DAT

Následující kapitola se věnuje sestavování množin trénovacích dat pro algoritmy učení transformačních slovníků. Trénovací data jsou algoritmu předána ve formě matice, jejíž sloupce tvoří úseky signálu, na který má být slovník optimalizován. Na základě velikosti segmentu, který zpracovává OMP bude odpovídající i velikost segmentu pro učení K-SVD slovníku.

### 6.1 Výběr z celého signálu

První a nejjednodušší možností jak vybírat data k sestavení trénovací matice je vzít rekonstruovaný signál a ten po segmentech poskládat do matice. Tato metoda ale není optimální. Například se u celé písně dá předpokládat několikanásobná změna celé scény. Tedy střídání refrénu, zpěvu a různých nástrojů v různých částech skladby. V takovém případě se může stát, že rekonstruovaný segment bude doplňovaný atomy, které nejsou optimalizované pro daný úsek.

Samotný výběr by pak prakticky neměl zahrnovat poškozené oblasti. Nemá smysl slovník optimalizovat na chyby. Proto se nějakým mechanismem musí tyto úseky odstranit. Nejjednodušší je díry prostě vystříhnout. Problém ale nastane při „slepování“ dané oblasti. Je velká pravděpodobnost, že vznikne nespojitost a ta zanechá do výsledného segmentu chybu. Řešením by bylo úseky vybírat právě tak, aby okraje díry byly zároveň okraje výběrového segmentu. Vhodnost takového výběru bude otestována v kapitole věnující se srovnání.

### 6.2 Výběr z lokální oblasti signálu

Dalším způsobem sběru trénovacích dat by bylo tyto segmenty poskládat ze stacionárních segmentů skladby. Tím se myslí jasně rozdělené oblasti přímo přilehlé k díře, případně oblasti s velkou podobností k této části. To znamená pokud se díra nachází uvnitř refrénu, bylo by vhodné trénovací data poskládat z refrénů v jiných místech skladby. Takový přístup dává celému algoritmu větší smysl. Problém může být množství dat potřebné k sestavení matice.

Takové vybírání může být určeno uživatelem, který podle poslechu podobné oblasti definuje. Nicméně aby se dal proces algoritmovat, diplomová práce se zaměří na určení a vybírání stacionárních segmentů z libovolného signálu s využitím algoritmu pro adaptivní segmentaci.



## 6.2.1 Segmentace na stacionární úseky

Pro rozdělení libovolného vstupního signálu na stacionární segmenty a jejich následné využití do trénovacích dat byl zkoumán a využit algoritmus popsáný v práci [16]. Autor zde popisuje metody – především několik modifikací algoritmu greedy selection. Tato metoda je aplikována na vstupní signál pro určení stacionárních úseků. Detailnímu popisu se věnuje samotná disertační práce [16], ve které jsou srozumitelně uvedeny výhody a nevýhody algoritmu.

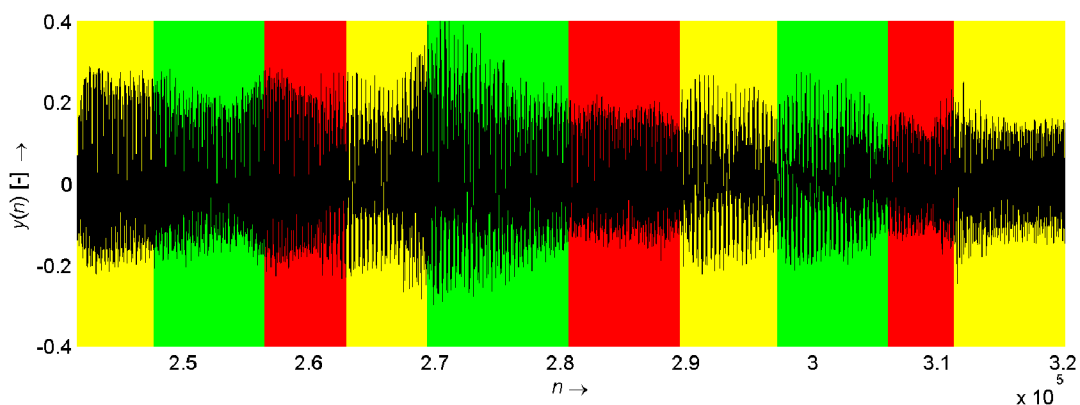
Klíčovým prvkem posuzování podobnosti dvou segmentů v algoritmu je tzv. koeficient lokální spektrální špičatosti (local spectral kurtosis). Jeho definice je

$$C(\mathbf{y}_{\text{seg}}) = \frac{\sum_{k=0}^{L-1} |\text{DFT}\{\mathbf{y}_{\text{seg}}(n)\}|^4}{\left(\sum_{k=0}^{L-1} |\text{DFT}\{\mathbf{y}_{\text{seg}}(n)\}|^2\right)^2}, \quad (6.1)$$

kde  $\mathbf{y}_{\text{seg}}$  je obecný segment (zatím neohodnocený) vybraný z obecného signálu  $\mathbf{y}(n)$ . Znak funkce  $\text{DFT}\{\}$  představuje diskrétní Fourierovu transformaci segmentu  $\mathbf{y}_{\text{seg}}$  o délce  $L$  jako

$$\text{DFT}\{\mathbf{y}_{\text{seg}}\} = \sum_{n=0}^{L-1} \mathbf{y}_{\text{seg}}(n) e^{-j2\pi kn/L}. \quad (6.2)$$

Funkcí algoritmu Greedy selection je rozdělit signál na úseky podle koeficientu  $C(\mathbf{y}_{\text{seg}})$ . Příklad rozdělení je vidět na obrázku 6.1. Barevně jsou zde odlišeny jednotlivé segmenty, které mohou být chápány právě jako lokální oblasti využitelné pro trénovací množiny.



Obr. 6.1: Segmentace na stacionární úseky pro hudební signál, žánr jazz

## 6.2.2 Aplikace segmentace pro výběr do trénovací množiny

Metoda greedy selection bude dále využita pro trénovací množinu. V předchozí části je popsáno, jakým způsobem se vytváří segmenty rozdělené do stacionárních úseků.

Nyní je nutné tyto segmenty ohodnotit. K tomu opět poslouží koeficient lokální spektrální špičatosti. Každý segment tedy ponese svou hodnotu, která jej charakterizuje. Předpoklad je, že dva charakterově podobné segmenty v různé části skladby budou mít velmi blízké koeficienty. Vhodnost tohoto koeficientu použitého jako kvantifikátor úseků se bude ověřovat v praktické části práce.

Signál je nejprve nutné předzpracovat. To znamená vyplnit díry nulovými hodnotami. Tento krok musí být proveden ještě před spuštěním algoritmu greedy selection. Důvodem je, že nuly v místě díry nijak neovlivní hodnotu koeficientu. Zároveň je možné detekovat konkrétní segment, který se nachází na dané díře a koeficient s ním spojený. V případě delší díry nebo výrazné změny charakteru signálu mohou být v daném místě 2 segmenty. Tyto segmenty budou z výsledné trénovací sady odstraněny, protože obsahují poškozená data. Jejich koeficienty se ale započítají.

V tabulce 6.1 je naznačen průběh výběru segmentů. Vstupními parametry algoritmu jsou segmenty signálu  $\mathbf{y}_i$ , jejich koeficienty  $\mathbf{c}$  a hodnota  $perc_t$ , která definuje uživatelem požadované procento vybraných dat ze vstupního signálu. V algoritmu se dále nachází proměnné  $d$  představující diferenci a  $\mathbf{t}$  – vektor který ponese vybraná trénovací data. Po inicializační části algoritmu se vybírají koeficienty z intervalu

Tab. 6.1: Algoritmus pro výběr segmentů podle koeficientů

<p><b>Vstupní parametry:</b> <math>\mathbf{y}_i, \mathbf{c} = [C(\mathbf{y}_1), C(\mathbf{y}_2), \dots, C(\mathbf{y}_L)], perc_t</math></p> <pre> <b>t</b> = 0; perc = 0; d = 0; <b>while</b> perc &lt; perc<sub>t</sub>   <b>for</b> i = 1 : L <b>do</b>     <b>if</b> {(C(<b>y</b><sub>i</sub>) &lt; C(<b>y</b><sub>d</sub>) + d) &amp;&amp; (C(<b>y</b><sub>i</sub>) &gt; C(<b>y</b><sub>d</sub>) - d)}       <b>t</b> += <b>y</b><sub>i</sub>;     <b>end</b>     perc = 100 · length(<b>t</b>) \ length(<b>y</b>);     d = d + 0,0001;   <b>end</b> <b>end</b> </pre>
<p><b>Výstupní parametr:</b> <b>t, perc</b></p>

$(C(\mathbf{y}_d) - d; C(\mathbf{y}_d) + d)$ . Všechny segmenty s  $C(\mathbf{y}_i)$  vyhovující podmínce se přidají do  $\mathbf{t}$ . Následně se spočítá, jaká poměrná část byla ze signálu vybrána a pokud výsledná hodnota přesáhne přednastavený práh  $perc_t$ , podmínka while cyklu algo-

ritmus ukončí. Pokud nebude vybráno dostatečné množství dat, zvýší se hodnota odchylky  $d$  o číslo 0,0001 (tím se ve výsledku zvýší i množství vybraných dat) a výběr se opakuje.

Hodnota 0,0001 pro navyšování  $d$  byla měřením a experimenty zvolena jako kompromis mezi dobou počítání algoritmu a procentuální odchylkou výběru. S vhodně nastavenou délkou okénkové funkce a délkou zpracovávaného segmentu pro OMP není třeba používat parametr „shift“ u K-SVD. To znamená, že pokud bude délka segmentu OMP 512 vzorků, okénko analýzy pro greedy selection bude také 512 vzorků.

## 7 IMPLEMENTACE A APLIKACE

Pro testování jsou k rekonstrukcím využity implementace algoritmů pro učení slovníků z toolboxu „SMALLbox v2.1“ [6]. Jedná se o robustní balík, který obsahuje dílčí toolboxy. Konkrétně algoritmus K-SVD je realizován v „KSVD-Box v13“. Algoritmus pro inkoherentní slovníky je v doplňkovém balíku „incoherentdl-v1\_1“. Samotný toolbox je volně dostupný na stránkách tvůrců na adrese: <http://small-project.eu/software-data/smallbox/>.

Problémem SMALLboxu je jeho roztržitost a špatná provázanost vnitřních funkcí. Ve verzi získané ze stránek tvůrců některé metody nefungují správně. Z toho důvodu je opravená verze přiložena k diplomové práci v elektronické podobě. Do této verze jsou také nainstalovány doplňkové balíčky, které v původním SMALLboxu nejsou a zároveň jsou odstraněny nevyužívané toolboxy.

Zatímco KSVD-Box má své výpočetně náročné funkce realizované v jazyce C za pomoci MEX souborů, inkoherentní algoritmy jsou programované pouze v Matlabovském kódu. To znamená značnou časovou nevyváženost. Zatímco K-SVD slovník o velikosti  $1024 \times 1024$  trvá naučit zhruba 10 minut, inkoherentní slovník aplikuje výpočet pár hodin při stejném množství trénovacích dat. Snížení množství trénovacích dat tyto doby redukuje. Nižší hodnoty nastavené koherence tyto doby zase prodlužuje. Z toho důvodu se praktické výsledky zaměří na doplňování menších děr a budou rekonstruovány kratšími segmenty. V případě inkoherentního slovníku to bude maximálně segment o velikosti 256 vzorků a slovník o velikosti  $256 \times 256$ .

Samotné rekonstrukce signálů jsou implementovány v Audio Inpainting toolboxu. Toolbox obsahuje sadu matlabovských funkcí vyvíjených v rámci projektu „Nové metody doplňování chybějících vzorků v audio datech“ na Fakultě elektrotechniky a komunikačních technologií VUT v Brně. Jeho představení se věnuje [12].

Všechny uvedené funkce potřebují pro svůj chod nainstalovaný Matlab ve verzi alespoň R2012a. Navíc většina dílčích funkcí v Audio Inpainting toolboxu má založené konstrukce základních operátorů na dalších toolboxech LTFAT (The Large Time-Frequency Analysis Toolbox) [17] a UnLocBox (Matlab convex optimization toolbox) [15]. Pro správnou funkčnost funkcí v následující kapitole je vyžadována instalace obou.

### 7.1 Popis funkcí

V elektronické příloze se nachází funkce a skripty potřebné pro spuštění experimentů. Inicializace celého rozhraní se provede spuštěním skriptu „run\_initialization.m“. Skripty pro hromadné testování se nacházejí ve složce „experimenty/“. Pro reprodukci výsledků dosažených v části 8.1.2 slouží skript:

- test\_na\_cele\_skladbe.m.

Zde se pro všechny nastavené žánry spustí učení slovníku metodou K-SVD, pro který se trénovací data sesbírají na celé skladbě. Výsledkem jsou rekonstruované hudební soubory s ohodnocenou kvalitou rekonstrukce parametrem SNR uloženým v textovém souboru.

Experimenty z kapitoly 8.1.3 lze opakovat ve skriptu:

- test\_stacionarni\_segmentace.m.

Spuštěním se postupně pro definované hodnoty volá funkce „stacionarni.m“. V ní je realizovaný algoritmus z části 6.2. Pro různé hodnoty výběru (procentuální část využití signálu k výběru) se ve funkci sestavuje množina trénovacích dat. Ta se následně využije k učení slovníku a následnou rekonstrukci signálů.

Ke spuštění hromadného testu, který pro všechny nastavené žánry spustí trénování inkoherentního slovníku a s ním postupně rekonstruuje jednotlivé skladby slouží skript:

- test\_koherence.m.

Jedná se o experiment popsáný v kapitole 8.2.

Poslední skript pro hromadné testování je

- test\_pro\_pemoq.m.

Jeho účelem je sesbírat trénovací data na celém signálu a následně ze signálu vybrat 4 vteřiny. Tato část se použije k rekonstrukci. Výsledky jsou poté hodnoceny PEMO-Q. U všech skriptů pro hromadné testování je navíc možné změnit slovník, se kterým se signál rekonstruuje. K tomu slouží parametr „script“ na začátku všech souborů.

## 7.2 Implementace PEMO-Q

Praktické experimenty využívají k hodnocení hudebních souborů kromě objektivní metody SNR také PEMO-Q. K výpočtu všech uvedených PEMO-Q parametrů byla v diplomové práci využita demo verze programu PEMO-Q Demo-Version 1.3 od Hör-Tech gGmbH. Jedná se o verzi volně ke stažení, ovšem s omezeními. Například pro každý testovaný soubor musí být opísován captcha kód a otestované soubory mohou být dlouhé pouze 4 sekundy. Tato verze je zdarma ke stažení na adrese autorů: [http://www.hoertech.de/web\\_en/produkte/downloads.shtml](http://www.hoertech.de/web_en/produkte/downloads.shtml).

## 7.3 Knihovna hudebních signálů

Pro měření a aplikace algoritmů byla využita databáze GTZAN. Jedná se o kolekci hudebních skladeb rozdělených do 10 hudebních žánrů. Konkrétně: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae a Rock. Pro každý žánr je

sestavena kolekce 100 písní přes široké spektrum interpretů. Každá nahrávka má následující parametry:

- Délka: 30 sekund,
- Kanály: Mono,
- Vzorkovací frekvence: 22050 Hz,
- Kódování vzorků: 16-bitů,
- Formát: .au.

Výhodou kolekce je rozsáhlost a písně navíc odpovídají svému přiřazení k žánrům. I přes stejné parametry nahrávky je nevýhodou jejich rozdílná kvalita. Některé nahrávky jsou kvalitnější, jiné obsahují výrazně slyšitelný šum. Pro testování jsou tedy z celé kolekce vybírány skladby s odpovídající si kvalitou na základě poslechu.

Kvůli velikosti a autorským právům databáze nejsou soubory přiložené k elektronické příloze práce. Pro reprodukci výsledků dosažených v práci je třeba databázi stáhnout ze stránek autora, soubory umístit do složky /experimenty/signal\_input/ a odkomentovat položky ve skriptech pro hromadné testování.

## 8 VÝSLEDKY A POROVNÁNÍ

Snahou diplomové práce je aplikovat algoritmus segmentace stacionárních úseků do trénovacího procesu. Dále pak ověřit použitelnost inkoherentního slovníku ve srovnání s algoritmem K-SVD a statickým DCT. Dále představené experimenty se zaměřují na srovnání účinnosti rekonstrukce pro různé žánry.

Kvalita rekonstrukce se bude hodnotit na základě vypočítaného SNR. Hodnota SNR nabízí slušnou představu o úspěšnosti rekonstrukce, ovšem výpočet neuvažuje vlastnosti lidského sluchového aparátu. Z toho důvodu se nakonec specificky upravené soubory ohodnotí pomocí metody PEMO-Q.

### 8.1 Hodnocení metod sběru dat

Následující podkapitoly se věnují sbírání trénovacích dat. Sestavení trénovací množiny je klíčová část učení slovníku. Snížením množství dat se jednak sníží doba trénování slovníku a zároveň se přesněji specifikují trénovací data.

#### 8.1.1 Aplikované sady děr

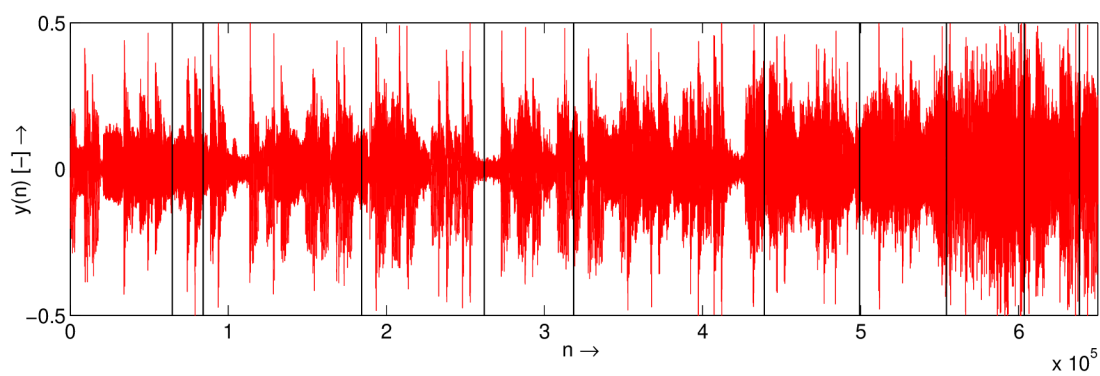
Pro účely měření kvality rekonstrukce byly navrženy dvě matice sady chybějících vzorků. Jedná se celkem o 10 shluků chybějících vzorků o různých délkách, které představují díry v signálu. Kvůli charakteru testů a srovnání účelnosti rekonstrukce byly sestaveny dvě sady. Jednak sada malých děr o velikosti mezi 0,8 – 3,4 ms, a sada větších o velikosti 2,0 – 13,2 ms. Konkrétní délky a pozice je možné vidět v tabulce 8.1, kde pole začátek označuje začátek poškozené oblasti,  $l$  délku oblasti a  $t$  dobu trvání díry v použitém vzorkovacím kmitočtu.

Nevýhodou sady malých děr je jejich délka. Lidské ucho není schopné detekovat díry pod 2 ms. Většinu z nich tedy nemá smysl rekonstruovat. Ale v rámci experimentů bude přesto pozorována kvalita jejich rekonstrukce. Důvodem je dlouhá výpočetní doba INK-SVD u větších segmentů při velkém množství trénovacích dat.

Pro lepší představu o rozložení děr v hudebních souborech dlouhých 30 vteřin jsou na obrázku 8.1 díry aplikovány. Na popové skladbě jsou černými čarami vidět poškozená místa, která budou rekonstruována.

Tab. 8.1: Seznam děr pro experimenty

Začátek [vzorků]	<i>Malé díry</i>		<i>Velké díry</i>	
	l [vzorků]	t [ms]	l [vzorků]	t [ms]
64524	43	2,0	43	2,0
84003	60	2,7	288	13,1
184228	32	1,5	127	5,8
261765	75	3,4	275	12,5
318309	22	1,0	241	10,9
438943	33	1,5	48	2,2
499185	17	0,8	292	13,2
554202	58	2,7	146	6,6
603395	39	1,8	238	10,8
638278	63	2,9	288	13,1



Obr. 8.1: Rozložení děr na popové skladbě



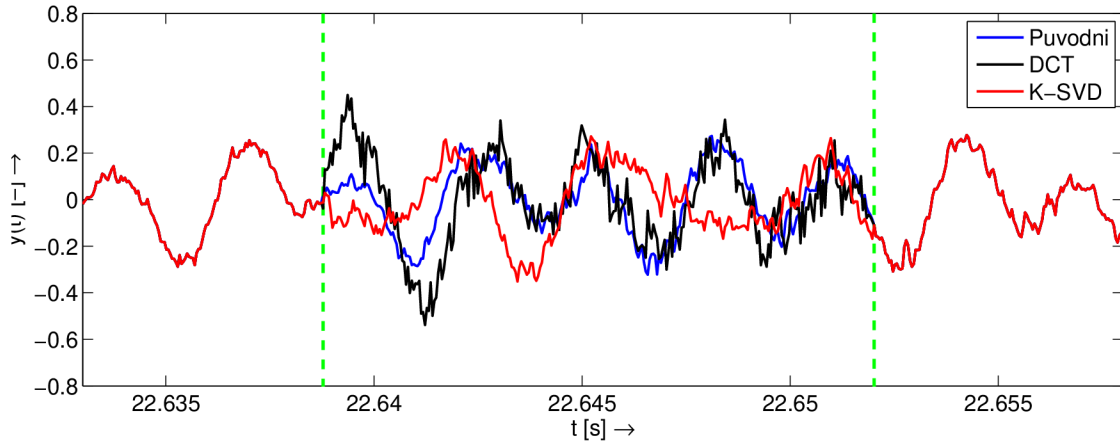
### 8.1.2 Experiment pro trénovací data z celé skladby

V části 6.1 byla představena metoda sběru trénovacích dat z celé skladby. Následující experiment tuto metodu aplikuje na učení slovníků. Všechny zpracovávané signály mají z důvodu zajištění stejných podmínek pro rekonstrukce délku 30 sekund. Jak bylo uvedeno, nahrávky mají vzorkovací frekvenci 22050 kHz.

Pro optimalizaci K-SVD slovníku byla sesbírána trénovací data z celého známého signálu. Tedy z rekonstruované skladby byly vymazány díry a poskládala se trénovací matice s parametrem „shift“ rovným délce zpracovávaného segmentu. Pro srovnání účinnosti jednotlivých slovníků byly rekonstruovány právě obě velikosti sad děr. Zároveň byly testy provedeny na všech 10 uvedených žánrech z databáze GTZAN. Segmenty předložené OMP jsou pro malou sadu děr 256 vzorků a pro velkou 1024 vzorků. Tabulky A.1 a A.2 přiložené v příloze ukazují výsledky. Zelenou barvou jsou vyznačeny lepší výsledky.

Jak je vidět v tabulkách, pro sadu malých děr jsou hodnoty SNR lepší u slovníku DCT. Rekonstrukce s K-SVD dopadla lépe pouze u žánrů Disco, Hiphop, Jazz a Rock. Ovšem u sady velkých děr už K-SVD jasně zvyšuje SNR.

Jako příklad je na obrázku 8.2 ukázán výsledek experimentu. Jedná se o skladbu žánru pop, a rekonstrukce jedné z větších děr.



Obr. 8.2: Příklad rekonstrukce skladby pop

Největším problémem použitého K-SVD slovníku mohou být právě trénovací data. K optimalizaci K-SVD je použita celá skladba, která má několik dynamicky se měnících scén. Může tedy nastat situace, kdy se bude díra umísťená v refrénu rekonstruovat za pomoci atomů, které byly optimalizované pro úplně jinou část signálu.

### 8.1.3 Experiment pro trénovací data ze stacionárních segmentů

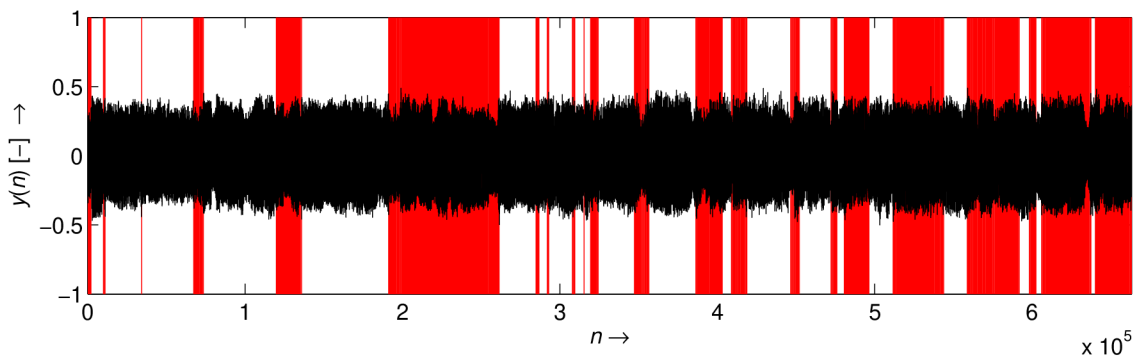
Následující část se věnuje využití algoritmu pro segmentaci stacionárních úseků. Jak bylo naznačeno v kapitole 6.2.2, vstupním parametrem metody je požadovaná poměrná část výběru segmentů ze signálu. Tedy při zadání  $perc_t = 20$  se ze vstupního signálu vybere alespoň 20% stacionárních segmentů, podobných jako v rekonstruovaných místech do trénovacích dat.

Na obě sady připravených děr byl aplikován experiment s proměnnou hodnotou  $perc_t$ . Pro hodnoty  $perc_t = 5 : 5 : 70$  byly postupně skládány trénovací matice, na které se natrénoval K-SVD slovník. Výsledný slovník byl využit k rekonstrukci obou sad děr. Pro každou dílčí rekonstrukci byla zaznamenána hodnota SNR.

V grafech na obrazcích B.1 a B.2 jsou výsledky pro malé díry, na B.3 a B.4 pro velké díry. Pro většinu žánrů lze pozorovat navýšení SNR při snížení množství vybraných trénovacích dat. To znamená že čím nižší procento výběru je nastaveno, tím specifičtější bude výsledný slovník optimalizován na oblasti signálu ve kterých se nachází díry.

U rekonstrukcí s velkou sadou děr je možné pozorovat stálejší průběhy a tedy nedochází zde k výraznému vychýlení hodnot jako u malých děr. Dalším rysem druhého páru grafů je délka křivek. Je například vidět, že u žánru Pop nebylo možné dosáhnout výběru pod 45 % a nad 71 %. To je důsledek odlišné délky úseků. Pokud bude mít signál dlouhé stacionární úseky a segmenty s dírami budou z trénovací množiny odstraněny, nebude možné vybrat data například nad 71 %. Zároveň jak lze odhadnout z principu algoritmu – pokud bude při dlouhých úsecích v jedné iteraci vybráno větší množství dat než je podmínkové  $perc_t$ , bude algoritmus ukončen. Ovšem díky délce segmentů může přiřazení pár segmentů do množiny tvořit velkou část signálu. Proto není například možné dosáhnout hodnoty pod 45 %.

Na obrázku 8.3 je vidět jaká data byla vybrána ze skladby žánru Country při  $perc_t = 30$ . Červeně označené oblasti jsou skládány do trénovací matice.



Obr. 8.3: Výběr 30 % stacionárních segmentů ze skladby žánru Country

## 8.2 Srovnání algoritmů pro učení slovníků

Pro účely srovnání inkohherentního slovníku s K-SVD algoritmem a DCT slovníkem byl sestaven následující experiment. V první řadě je pozorován vliv nastavené maximální vzájemné koherence na účinnost rekonstrukce signálu u INK-SVD. Vzájemná koherence může nabývat hodnot mezi  $\langle 0, 1 \rangle$ . Výsledky rekonstrukce pro všechny žánry jsou vyneseny do grafů na obrázcích B.5 a B.6. Hodnota koherence se postupně mění po kroku 0,1. Jak je vidět na výsledku, pro většinu žánrů má průběh rostoucí charakter. Tedy s klesající vzájemnou koherencí se zvyšuje SNR, ovšem metoda je podstatně závislá na dekorelovaném signálu. Díky tomu lze pozorovat výkyvy hodnot. Například jak je patrné v grafu na hodnotách 0,7 a 0,8.

Obecně lze tedy říct, že použití inkohherentního slovníku místo obyčejného K-SVD má smysl. Nelze ale jednoznačně určit obecnou hodnotu vzájemné koherence, pro kterou budou rekonstrukce nejlepší u všech žánrů. Nadruhou stranu  $\mu_0$  nastavené mezi 0,1 a 0,2 vykazuje alespoň srovnatelné nebo lepší hodnoty SNR oproti K-SVD. Přehled nelepších hodnot z konfigurací je uveden v tabulce A.3. Jedná se o sadu malých děr. Pro K-SVD byl použit algoritmus pro stacionární segmentaci a vybíralo se nejvyšší SNR z experimentu.

Jak ukazují výsledky, ve srovnání tabulek A.1 a A.3 je možné vidět pozitivní vliv aplikace algoritmu pro segmentaci stacionárních úseků. Stejně tak zavedení dekorelačního kroku pozitivně ovlivní výsledky pro část žánrů.

## 8.3 Hodnocení na základě PEMO-Q

Jak bylo uvedeno v části popisující testovací rozhraní metody PEMO-Q, uvedená demoverze zpracuje signály dlouhé maximálně 4 sekundy. Z toho důvodu byly předchozí experimenty upraveny. K testům je tentokrát použita sada velkých děr, které jsou rozmístěny v 4 sekundovém úseku vybraného z původního signálu. Malé díry zde nemá smysl rekonstruovat, protože metoda PEMO-Q není dostatečně citlivá a výsledky by byly totožné. Tedy signály by byly označeny za téměř shodné.

Byla provedena rekonstrukce se slovníky DCT, K-SVD a nakonec INK-SVD. Pro trénovací algoritmy se vybrala trénovací data z celého 30 sekundového signálu s použitím algoritmu pro výběr stacionárních segmentů. Parametr  $perc_t$  měl hodnotu 20. Hodnota maximální vzájemné koherence pro INK-SVD byla nastavena na 0,2.

Účelem experimentu je ohodnocení kvality rekonstrukce druhou metodou PEMO-Q pro různé žánry a slovníky. Výsledky jsou uvedeny v tabulce A.4. Ve srovnání je použita také hodnota SNR. Výsledkem PEMO-Q metody jsou parametry PSM a PSMt. Pro zavedený experiment nemá smysl pozorovat parametr PSM, protože téměř pro každý žánr je jeho hodnota 0,999.

Pro lepší přehlednost jsou nejvyšší hodnoty označeny zelenou a nejnižší červenou barvou. Z výsledků jsou patrné protiklady. Například u DCT slovníku jsou pro žánry Blues, Country a Hiphop hodnoty SNR nejnižší ze všech slovníků, hodnoty PSMt jsou ale nejvyšší. Zvláštní jev je způsoben odlišným principem hodnotících metod. Zatímco SNR vyhodnocuje podobnost signálů, PEMO-Q vyhodnocuje kvalitu rekonstrukce pro lidské sluchové ústrojí.

## 8.4 Poslechové testy

Rekonstruované skladby byly nakonec podrobeny poslechovému testu. Jak ukazují výsledky SNR oproti PEMO-Q, kvalita rekonstrukce může být na poslech horší i při vyšší hodnotě SNR. V signálech, u nichž byla rekonstruována sada malých děr mají na poslech velmi dobrou kvalitu. Díry kratší než 2 ms jsou neslyšitelné. Díry nad 2 ms už slyšitelné jsou ale ve většině skladeb se ztrácí. Ve skladbách žánru classical, disco, hiphop, jazz, metal, pop, reggae a rock jsou pak všechny téměř nepostřehnutelné. V žánrech blues a country jsou delší úseky slyšet ale při poslechu neruší. Při doplnění děr zmíněnými metodami se jejich polechovost zlepšuje. Obzvláště se slovníkem DCT jsou oblasti téměř nezaznamenané.

Pro signály s doplněnými delšími dírami jsou již slyšitelné výraznější rozdíly. Pokud jsou chybějící shluky vzorků v signálu ponechány je v nich slyšet nepříjemné praskání. To platí pro všechny žánry obdobně. Nejlepším dojmem působí rekonstrukce s DCT slovníkem. Doplněné díry jsou méně slyšitelné navíc v signálu nepůsobí rušivě a tlumí se zmíněné praskání, které tvoří vynechané vzorky. Rekonstrukce s K-SVD slovníkem má na většinu dopočítaných úseků kladný vliv. Většinou poškozené oblasti působí stejně jako u DCT ale zvláště u těch delších je výrazněji slyšitelný šum.

S využitím algoritmu pro inkoherentní slovníky jsou zase u většiny signálů opravené úseky na poslech nepříjemné. Díky úpravám atomů kvůli zajištění inkoherece se mění podoba celého slovníku a oproti K-SVD přináší tyto rekonstrukce mnohem výraznější hladinu šumu. I při menších opravených úsecích je slyšet výrazné a nepříjemné šumění. Tedy zatímco s ohodnocením SNR přináší INK-SVD zlepšení, na poslech přináší spíše zhoršení.

## 9 ZÁVĚR

Úkolem diplomové práce bylo využít algoritmus pro segmentaci stacionárních úseků, použít ho pro učení transformačních slovníků a ten následně aplikovat společně s metodou OMP k doplňování chybějících oblastí ve zvukovém signálu. Jako zástupce algoritmu pro učení slovníku byl zvolen K-SVD a jako zástupce inkohherentního slovníku byl vybrán INK-SVD.

První kapitola se věnuje teorii řídkých reprezentací. Následující část pak teoreticky popisuje rekonstrukce zvuku. Třetí kapitola dodává přehled mezi různými typy slovníků – statické a optimalizované trénovacím algoritmem. Následující část detailně vysvětluje K-SVD algoritmus a věnuje se nastavení jeho základních parametrů. Především pak optimální a obecné konfiguraci těch nejdůležitějších. Přidáním kroku dekorelace atomů do K-SVD je dosaženo natrénování inkohherentního slovníku. Tato modifikovaná metoda se jmenuje INK-SVD. Obecným popisem metody a vysvětlením dekorelačního kroku se zabývá pátá kapitola práce.

Metody sběru trénovacích dat jsou popsány v šesté části. Především je zde uveden návrh na využití adaptivní segmentace pro sběr trénovacích dat. S jeho využitím lze získat charakterově podobné části skladby do trénovací množiny. Tedy například pro díru umístěnou v refrénu skladby se signál rozdělí na charakterově odlišné segmenty a vyberou se ty nejbližší k segmentu s chybějícími vzorky. To mohou být právě refrény v jiných místech skladby a další podobné úseky.

Pro výběr stacionárních segmentů byla v prostředí Matlab vytvořena funkce. Navíc byly sestaveny skripty pro hromadné testování rekonstrukce metodou OMP s využitím uvedených transformačních slovníků. K testování jsou využity sady malých a velkých děr. Výsledky hromadných testů ukazují, že hodnoty SNR jsou s použitím K-SVD algoritmu vyšší než při DCT a pro většinu žánrových skladeb má smysl slovník trénovat. Dále byl pozorován vliv výběru trénovacích dat z lokálních oblastí v porovnání se sbíráním dat z celé skladby. Bylo zjištěno, že vybírání pouze určitého procenta segmentů do trénovací množiny na základě koeficientu lokální spektrální špičatosti pozitivně ovlivní rekonstrukci. V poslední řadě se sledoval vliv snížení vzájemné koherence atomů slovníku na hodnotu SNR všech žánrů. Především snížení povolené maximální koherence na 0,1 a 0,2 se projevilo nárůstem SNR.

Rekonstruované signály byly navíc ohodnoceny metodou PEMO-Q a poslecho-  
vým testem a z výsledků jsou patrné rozpory ve vyhodnocovacích metodách. Zatímco SNR se s využitím komplexnějších a náročnějších metod pro učení slovníků zvyšovalo, degradovala se zároveň poslechovost doplňovaných signálů. Především pak inkohherentní slovníky přinášejí do rekonstruovaného signálu značné množství šumu.

## LITERATURA

- [1] ADLER, A.; EMIYA, V.; JAFARI, M.; aj. *Audio Inpainting*. *IEEE Transactions on Audio, Speech, and Language Processing*, ročník 20, č. 3, March 2012: s. 922–932, ISSN 1558-7916, doi:10.1109/TASL.2011.2168211.
- [2] AHARON, M.; ELAD, M. a A. BRUCKSTEIN. *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation*. *IEEE Transactions on Signal Processing*, 2006, vol. 54, issue 11, s. 4311-4322. DOI: 10.1109/TSP.2006.881199.
- [3] BARCHIESI, D. *Sparse Approximation and Dictionary Learning with Applications to Audio Signals*. Londýn, 2013. Disertační práce. Queen Mary, University of London.
- [4] BRUCKSTEIN, A. M.; DONOHO D. L. a M. ELAD. *From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images*. SIAM REVIEW, Society for Industrial and Applied Mathematics, Vol. 51 (2009), No. 1, pp. 34-81, ISSN 0036-1445.
- [5] CHEN, S. S.; DONOHO, D. L. a M. A. SAUNDERS. *Atomic Decomposition by Basis Pursuit*. *SIAM Journal on Scientific Computing* [online]. 1998, vol. 20, issue 1, s. 33-61 [cit. 2014-05-27]. DOI: 10.1137/S1064827596304010. Dostupné z: <<http://epubs.siam.org/doi/abs/10.1137/S1064827596304010>>
- [6] DAMNJANOVIC I.; DAVIES, M. E. P. a M. D. PLUMBLEY. *SMALLbox - an evaluation framework for sparse representations and dictionary learning algorithms*, in Proc. LVA/ICA'10, [online], 2010, pp. 418–425, [cit. 2014-05-19]. Dostupné z: <<http://code.soundsoftware.ac.uk/projects/smallbox>>.
- [7] ELAD, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer Science+Business Media, 2010, 376 s. ISBN 9781441970107.
- [8] GOLUB, G. H.; Ch. F. V. LOAN. *Matrix computations: Johns Hopkins series in the mathematical sciences*. 2. vyd. Michiganská univerzita: Johns Hopkins University Press, 1989, 642 s. ISBN 0801837723.
- [9] HUBER, R.; B. KOLLMEIER. *PEMO-Q: A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception*. *IEEE Transactions on Audio, Speech and Language Processing*. 2006, vol. 14, issue 6, s. 1902-1911. DOI: 10.1109/TASL.2006.883259.

- [10] HRBÁČEK, R.; RAJMÍČ, P.; VESELÝ, V. a J. ŠPIŘÍK. *Řídké reprezentace signálů: úvod do problematiky*. Elektrovue [online]. 2011, [cit. 2014-05-15]. Dostupné z: <<http://elektrovue.cz/cz/download/ridke-reprezentace-signalu-uvod-do-problematiky/>>.
- [11] JAN, J. *Číslíková filtrace, analýza a restaurace signálů*. 2. rozšířené vydání. Brno: Nakladatelství VUTIUM, 2002. 427 s. ISBN 80-214-1558-4.
- [12] MACH, V.; WIESMEYR, C. *Methods of Completing Missing Samples in Audio Signal*. Conference Proceedings: International Masaryk Conference for Ph.D. Students and Young Researchers 2013. Hradec Králové: MAGNANIMITAS, 2013. ISBN: 978-80-87952-00-9.
- [13] MACH, V.; OZDOBINSKI, R. *Optimizing dictionary learning parameters for solving Audio Inpainting problem*. International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, 2013, roč. 2, č. 1, s. 40-45. ISSN: 1805- 5443.
- [14] MAILHÉ, B.; D. BARCHIESI a M. D. PLUMBLEY. *INK-SVD: Learning incoherent dictionaries for sparse representations*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2012, str. 3573–3576.
- [15] PERRAUDIN, N.; SHUMAN, D.; PUY G. a P. VANDERGHEYNST. *UNLoc-BoX A matlab convex optimization toolbox using proximal splitting methods*: ArXiv e-prints. [online]. roč. 2014, 1402.0779 [cit. 2014-05-23]. Dostupné z: <<http://arxiv-web3.library.cornell.edu/abs/1402.0779>>
- [16] RÁŠO, O. *Objektivní měření a potlačování šumu v hudebním signálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 148 s. Vedoucí disertační práce Ing. Miroslav Balík, Ph.D.
- [17] SØNDERGAARD, P. L.; TORRÉSANI B a P. BALAZS. *The Linear Time-Frequency Analysis Toolbox*. International Journal of Wavelets, Multiresolution Analysis and Information Processing. [online]. 2012, 10(4) [cit. 2014-05-23].
- [18] TROPP, J.A. *Greed is Good: Algorithmic Results for Sparse Approximation*. IEEE Transactions on Information Theory [online]. 2004, vol. 50, issue 10, s. 2231-2242 [cit. 2014-05-15]. DOI: 10.1109/TIT.2004.834793. Dostupné z: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1337101>>.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BP Basis Pursuit

DCT Diskrétní kosinová transformace – Discrete Cosine Transform

DFT Diskrétní Fourierova transformace – Discrete Fourier Transform

GLA Generalized Lloyd Algorithm

INK-SVD Inkoherentní K-SVD – Incoherent K-SVD

IPR Iterativní projekce a rotace

IRLS Iterative Reweighted Least Squares

LARS Least Angle Regression homothropy method

LS-OMP Least Squares Orthogonal Matching Pursuit

LTFAT The Large Time-Frequency Analysis Toolbox

OMP Orthogonal Matching Pursuit

PSM Perceptual Similarity Measure

PSMt Temporary Perceptual Similarity Measure

RLS-DLA Recursive Least Squares Dictionary Learning Algorithm

SNR Odstup signálu od šumu – Signal-to-Noise Ratio

SVD Singulární rozklad – Singular Value Decomposition

SR Řídké reprezentace – Sparse Representation

Weak-MP Weak Matching Pursuit



# SEZNAM PŘÍLOH

A Tabulky	50
B Grafy s výsledky rekonstrukce	53
C Obsah elektronické přílohy	56

## A TABULKY

Tab. A.1: Srovnání kvality rekonstrukce na celé skladbě – sada malých děr

<b>Slovník</b>	DCT	K-SVD
<b>Hodnocení</b>	SNR [dB]	SNR [dB]
Blues	28,892	22,254
Classical	29,966	26,726
Country	30,671	24,728
Disco	30,671	31,358
Hiphop	22,062	27,075
Jazz	28,986	29,405
Metal	25,409	25,095
Pop	28,828	28,058
Reggae	32,522	25,781
Rock	24,938	28,976

Tab. A.2: Srovnání kvality rekonstrukce na celé skladbě – sada velkých děr

<b>Slovník</b>	DCT	K-SVD
<b>Hodnocení</b>	SNR [dB]	SNR [dB]
Blues	24,434	24,540
Classical	20,388	27,030
Country	25,010	27,207
Disco	33,166	35,051
Hiphop	20,890	23,129
Jazz	31,515	30,222
Metal	22,155	22,871
Pop	28,102	33,146
Reggae	19,780	22,126
Rock	21,646	26,545

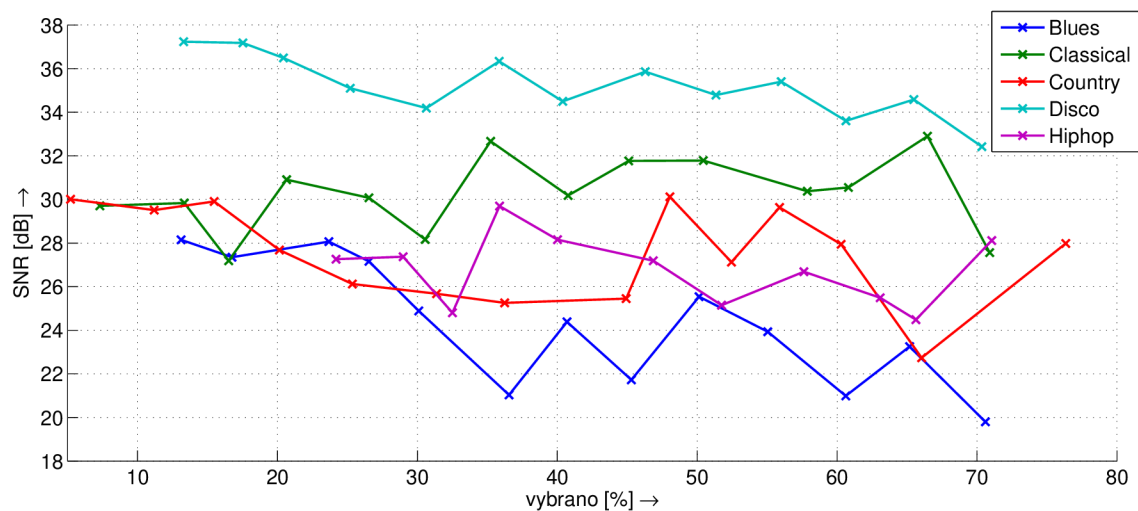
Tab. A.3: Srovnání rekonstrukce pro nejlepší parametry trénovacích algoritmů– sada malých děr

<b>Slovník</b>	DCT	K-SVD		INK-SVD	
<b>Hodnocení</b>	SNR [dB]	SNR [dB]	Vybráno [%]	SNR [dB]	$\mu_0$ [-]
Blues	28,892	28,145	13,113	32,387	0,1
Classical	29,966	32,902	66,457	30,834	0,1
Country	30,671	30,117	48,051	28,541	0,2
Disco	30,671	37,231	13,298	36,548	0,1
Hiphop	22,062	29,694	35,875	28,511	0,1
Jazz	28,986	32,234	17,601	36,669	0,2
Metal	25,409	28,584	20,008	27,546	0,2
Pop	28,828	34,048	35,255	34,947	0,1
Reggae	32,522	34,433	45,375	34,131	0,8
Rock	24,938	29,197	66,418	32,123	0,8

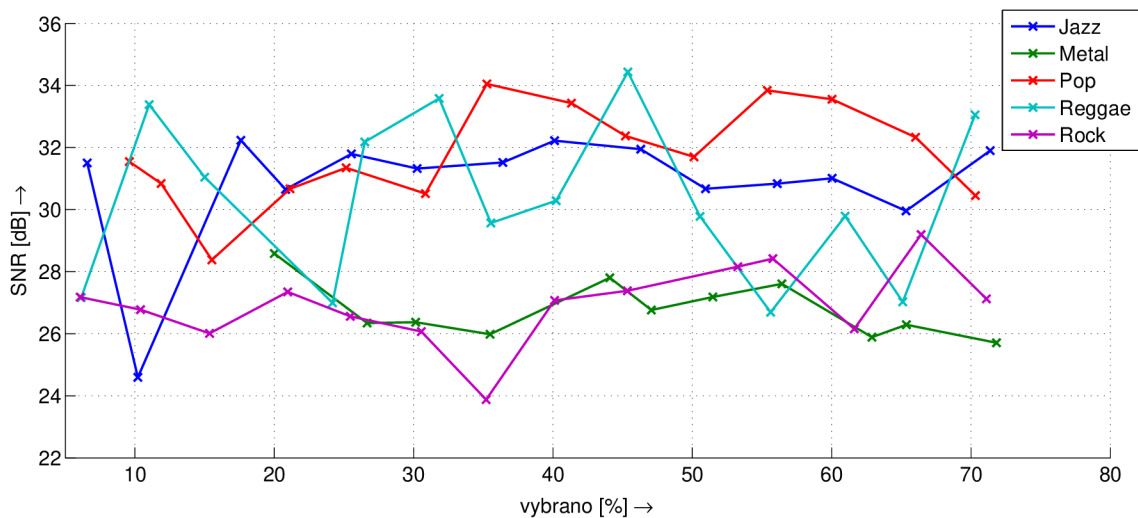
Tab. A.4: Srovnání rekonstrukcí pro různé slovníky s metodou PEMO-Q na sadě velkých děr

Slovník	DCT		K-SVD		INK-SVD	
	SNR [dB]	PSMt [-]	SNR [dB]	PSMt [-]	SNR [dB]	PSMt [-]
Blues	23,869	0,793	25,151	0,772	28,320	0,715
Classical	18,327	0,720	17,901	0,834	18,239	0,736
Country	10,836	0,772	16,727	0,754	15,603	0,705
Disco	20,674	0,820	19,920	0,795	20,104	0,810
Hiphop	10,446	0,787	16,962	0,786	15,730	0,665
Jazz	16,199	0,735	16,081	0,685	14,553	0,718
Metal	12,324	0,818	14,711	0,858	14,990	0,839
Pop	19,397	0,813	21,723	0,795	18,601	0,728
Reggae	17,102	0,823	20,188	0,842	19,578	0,777
Rock	12,306	0,770	16,842	0,828	18,565	0,734

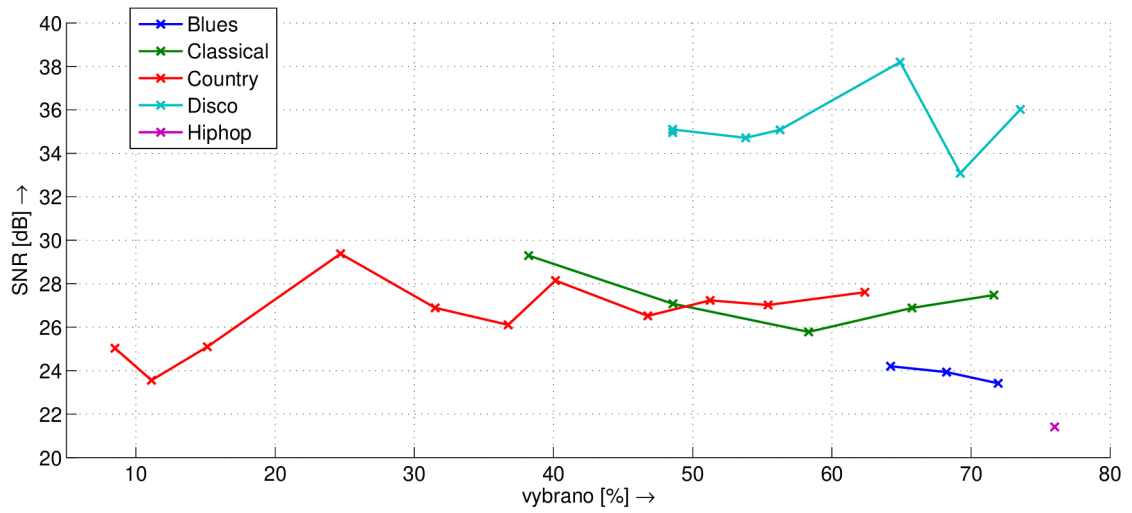
## B GRAFY S VÝSLEDKY REKONSTRUKCE



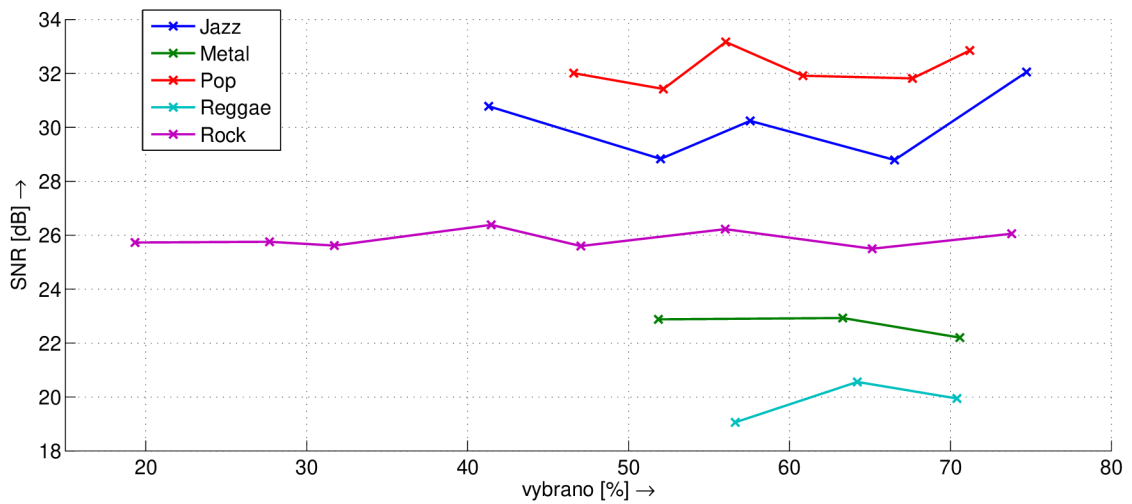
Obr. B.1: Výsledky rekonstrukce na malé sadě děr pro první polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD



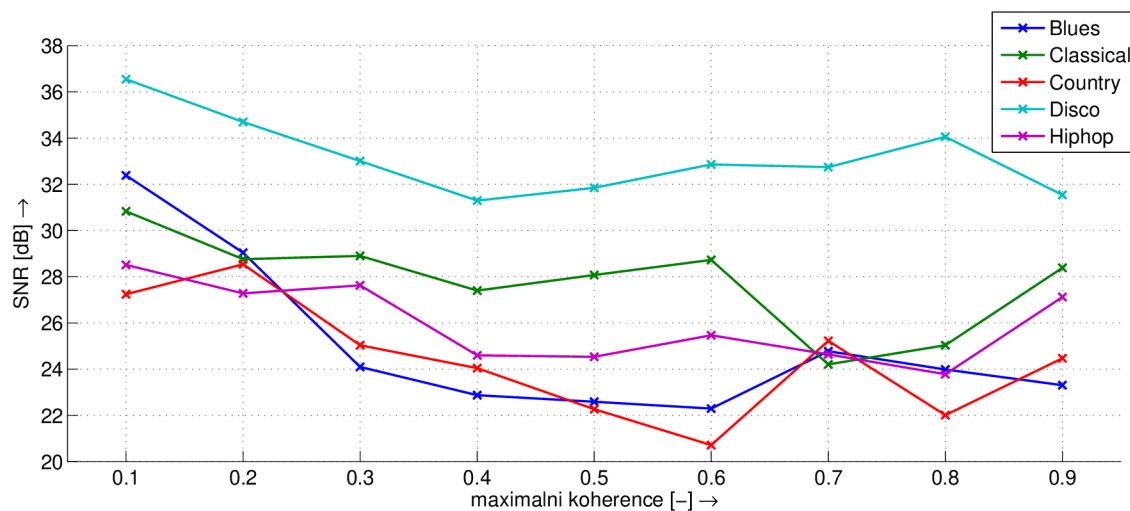
Obr. B.2: Výsledky rekonstrukce na malé sadě děr pro druhou polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD



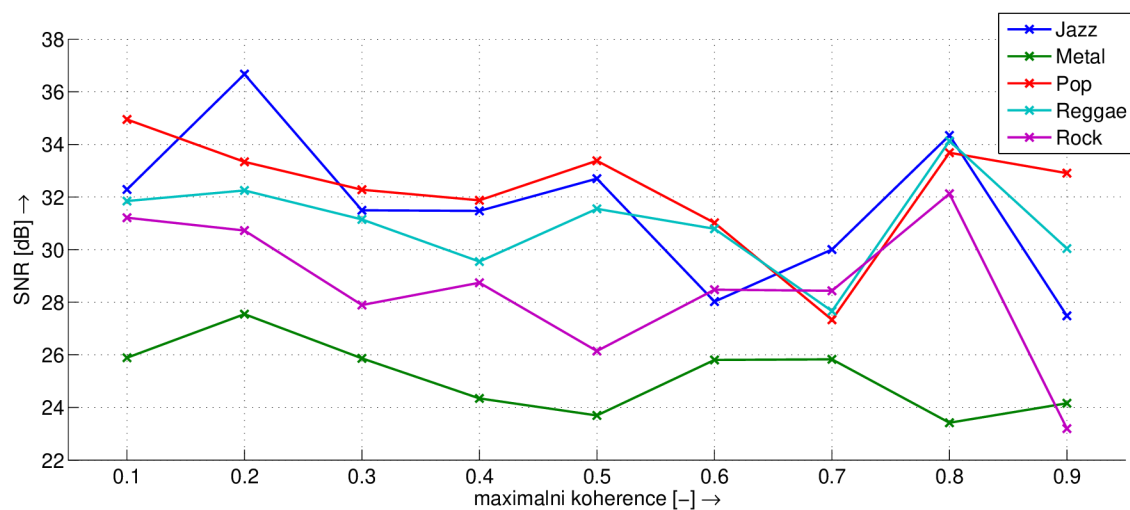
Obr. B.3: Výsledky rekonstrukce na velké sadě děr pro první polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD



Obr. B.4: Výsledky rekonstrukce na velké sadě děr pro druhou polovinu žánrů, závislost SNR na množství vybraných trénovacích dat pro K-SVD



Obr. B.5: Srovnání první poloviny žánrů, vztah SNR pro různé úrovně maximální koherence u INK-SVD



Obr. B.6: Srovnání druhé poloviny žánrů, vztah SNR pro různé úrovně maximální koherence u INK-SVD

## C OBSAH ELEKTRONICKÉ PŘÍLOHY

Na přiloženém CD je elektronická verze diplomové práce. Dále jsou na disku umístěny zdrojové kódy skriptů pro hromadné testování. Příloha obsahuje také upravenou verzi toolboxu SMALLBox. Struktura důležitých adresářů a jejich obsah je následující:

/smallbox-2.1/	– obsahuje SMALLbox
toolboxes/incoherentdl-v1_1/	– funkce pro učení inkoherentních slovníků
toolboxes/KSVD/	– funkce pro učení K-SVD
/experimenty/	– experimenty pro hromadné testování
signal_input/	– testované signály
signal_output/	– výsledky testování
gaps/	– definované sady děr
/ait/	– Audio Inpainting toolbox
solvers/	– funkce pro rekonstrukce (OMP)
scripts/	– skripty s parametry pro slovníky a OMP