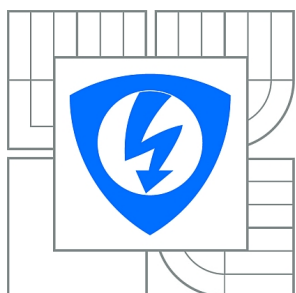


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

UČENÍ SLOVNÍKU PRO REKONSTRUKCI ŘÍDKÉHO SIGNÁLU

DICTIONARY LEARNING FOR SPARSE SIGNAL RECONSTRUCTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ROMAN OZDOBINSKI

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV MACH

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Roman Ozdobinski

ID: 125577

Ročník: 3

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Učení slovníku pro rekonstrukci řídkého signálu

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte techniky pro optimalizaci slovníku sloužícího k rekonstrukci řídkého signálu. Alespoň jednu techniku vyberte, realizujte v prostředí MATLAB a porovnejte objektivními metodami rekonstrukci přizpůsobeného slovníku s rekonstrukcí pomocí slovníků statických (Gaborův, DCT). Pro testování použijte různé typy zvukového signálu (hudební, řečový), kde budete rekonstruovat chybějící úseky.

DOPORUČENÁ LITERATURA:

[1] RAVELLI, Emmanuel, Gal RICHARD a Laurent DAUDET. Union of MDCT Bases for Audio Coding. IEEE Transactions on Audio, Speech, and Language Processing [online]. 2008, roč. 16(č. 8), 1361-1372 [cit. 2012-02-09]. DOI: 10.1109/TASL.2008.2004290. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4648210>

[2] HRBÁČEK, Radek, et al. Řídké reprezentace signálů: úvod do problematiky. Elektrevue [online]. 2011, 1, [cit. 2011-10-11]. Dostupný z WWW: <http://www.elektrevue.cz/cz/download/ridke-reprezentace-signalu--uvod-do-problematiky/>.

[3] ELAD, Michael. Sparse and Redundant Representations : From Theory to Applications in Signal and Image Processing. New York : Springer Science+Bussines Media, 2010. 376 s. ISBN 9781441970107.

Termín zadání: 6.2.2012

Termín odevzdání: 31.5.2012

Vedoucí práce: Ing. Václav Mach

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

ABSTRAKT

Tato bakalářská práce se zabývá učením slovníku pro rekonstrukci signálu na základě řídkých reprezentací. Jsou zde popsány způsoby tvorby statických a přizpůsobených slovníkových matic, které jsou užité s aproximativním algoritmem Orthogonal Matching Pursuit k rekonstrukci skupiny chybějících vzorků ve zvukovém signálu. Je zde teoreticky rozebrán algoritmus pro učení K-SVD slovníku společně s jeho implementací v prostředí Matlab. Dále jsou vybrané slovníky porovnávány na různých typech zvukového signálu.

KLÍČOVÁ SLOVA

K-SVD, řídké reprezentace, učení slovníku, OMP, rekonstrukce, zvuk

ABSTRACT

This bachelor thesis discusses the dictionary learning for the reconstruction of signal based on sparse representations. There are methods of static and optimized dictionary of matrices described that are used with approximative Orthogonal Matching Pursuit algorithm to reconstruct the missing groups of samples in the audio signal. There is theoretically analyzed algorithm for learning K-SVD dictionary together with its implementation in Matlab. Furthermore, the selected dictionaries are compared to the various types of audio signals.

KEYWORDS

K-SVD, sparse representation, dictionary learning, OMP, reconstruction, audio

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Učení slovníku pro rekonstrukci řídkého signálu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 především vedoucímu bakalářské práce panu Ing. Václavu Machovi za příkladné odborné vedení, věcné konzultace, cenné rady a podnětné návrhy k práci. Dále děkuji jeho kolegům z kanceláře PA-228 za občasně rady a v neposlední řadě své rodině za podporu.

Brno

.....

(podpis autora)

OBSAH

| | |
|--|-----------|
| Úvod | 10 |
| 1 Řídké reprezentace signálu | 11 |
| 1.1 Definice základních pojmů | 11 |
| 1.2 Řídká řešení a podmínky | 12 |
| 2 Rekonstrukce audio signálu | 13 |
| 2.1 Označení poškozených vzorků | 13 |
| 2.2 Segmentové zpracování | 14 |
| 2.3 Řídké reprezentace pro rekonstrukci zvukových segmentů | 15 |
| 2.4 Aproximační algoritmy | 15 |
| 2.4.1 Algoritmus OMP | 16 |
| 2.5 Výpočet SNR | 17 |
| 2.6 Výpočet RMSE | 18 |
| 3 Slovníky | 19 |
| 3.1 Statické slovníky | 19 |
| 3.1.1 DCT slovník | 19 |
| 3.1.2 Gaborův slovník | 20 |
| 3.2 Přizpůsobené slovníky | 20 |
| 3.2.1 Sjednocené MDCT báze | 20 |
| 3.2.2 K-SVD | 21 |
| 3.2.3 RLS-DLA | 21 |
| 3.2.4 ODL | 22 |
| 4 Natrénování K-SVD | 23 |
| 4.1 Kvantizace vektorů a K-means | 23 |
| 4.2 Popis algoritmu | 23 |
| 4.2.1 Použití K-means | 23 |
| 4.2.2 Použití SVD | 25 |
| 4.3 Sada vzorků | 27 |
| 4.3.1 Sestavení vzorků | 29 |
| 5 Implementace v prostředí matlab | 30 |
| 5.1 Audio Inpainting Toolbox | 30 |
| 5.1.1 Metody řešení | 30 |
| 5.1.2 Experimenty | 31 |
| 5.2 K-SVD toolbox | 31 |

| | | |
|----------|--|-----------|
| 5.2.1 | Práce s toolboxem | 31 |
| 6 | Parametry nastavení K-SVD | 33 |
| 6.1 | Volba posuvu segmentů | 33 |
| 6.2 | Volba inicializačního slovníku | 34 |
| 6.3 | Volba řídkosti | 34 |
| 6.4 | Volba počtu iterací | 35 |
| 7 | Srovnání slovníků | 37 |
| 7.1 | Průběh experimentu | 37 |
| 7.2 | Výsledky srovnání | 38 |
| 8 | Závěr | 41 |
| | Literatura | 42 |
| | Seznam symbolů, veličin a zkratk | 44 |
| | Seznam příloh | 45 |
| A | Obsah elektronické přílohy | 46 |

SEZNAM OBRÁZKŮ

| | | |
|-----|---|----|
| 4.1 | Původní signál | 27 |
| 4.2 | Poškozený signál | 28 |
| 4.3 | Trénovací signál | 28 |
| 6.1 | Srovnání posuvů | 33 |
| 6.2 | Srovnání inicializačních slovníků | 34 |
| 6.3 | Srovnání řídkosti | 35 |
| 6.4 | Průběh pro 200 iterací | 36 |
| 7.1 | Ženský hlas | 38 |
| 7.2 | Mužský hlas | 39 |
| 7.3 | Ženský zpěv | 39 |
| 7.4 | Hra na bicí | 40 |
| 7.5 | Hra na kytaru | 40 |

SEZNAM TABULEK

| | | |
|-----|--|----|
| 2.1 | Algoritmus OMP | 17 |
| 4.1 | Shrnutí K-SVD algoritmu | 26 |
| 5.1 | Parametry InpaintingExperiment | 31 |
| 5.2 | Parametry K-SVD | 32 |
| 7.1 | Nastavení experimentu | 37 |

ÚVOD

Libovolné vlnění je zpravidla ovlivněno přenosovým kanálem, kterým se šíří. V případě přenosu může nastat situace, kdy kanál nebude schopen přenést celou vlnu v původní podobě a bez poškození. Například výpadky v přenosu nebo nějakým dalším rušením. Tyto chyby mohou nastat v časové oblasti, tedy budou chybět části signálu nebo v oblasti úrovně a přenosový kanál pojme jenom část vlny – vznikne přebuzený signál.

Rekonstrukcí zvukového signálu se rozumí oprava průběhu ideálně do co největší podoby se signálem před jeho poškozením. Algoritmy řešící rekonstrukci průběhu na základě řídkých reprezentací jsou v zásadě aproximační. Každý má své výhody či nevýhody a jejich vlastnosti jsou silně závislé na parametrech, které jim uživatel předloží. Jedním z nejdůležitějších parametrů je tzv. slovník. Jedná se o matici, pomocí které je možné v kombinaci s řídkým vektorem aproximovat libovolný signál.

Tato bakalářská práce se zaměřuje na zhodnocení několika známých slovníků a algoritmů založených na řídkých reprezentacích, řešící rekonstrukce zvukového signálu, popis a význam jednotlivých parametrů a vhodnost jejich nastavení. První část práce teoreticky popisuje pojmy a podmínky přípustných řešení související s řídkými reprezentacemi. V následující kapitole jsou tato zjištění aplikována na zobrazené algoritmy spolu s dalšími definicemi jako násobící okénkové funkce a způsoby jak vůbec se signálem pracovat. Třetí část se věnuje pouze slovníkům. Jsou zde popsány především rozdíly mezi zástupci statických a přizpůsobených slovníků. Čtvrtá kapitola se věnuje algoritmu pro učení přizpůsobeného K-SVD slovníku. Dále pokračuje praktická část, kde se nejdříve popisuje implementace celého problému rekonstrukce v prostředí Matlab, následně se rozebírá vhodné nastavení K-SVD algoritmu a nakonec v poslední kapitole srovnání vybraných slovníků na různých druzích zvukového signálu.

1 ŘÍDKÉ REPREZENTACE SIGNÁLU

Řídké reprezentace (Sparse Representation – SR) úzce souvisí s nedourčeným systémem lineárních rovnic. Jedná se o soustavu, ve které je více neznámých než rovnic. Protože má soustava nekonečný počet řešení, různé metody založené na řídkých reprezentacích budou hledat taková, která mají nejvíce současně nulových koeficientů, respektive nejméně nenulových koeficientů. Právě tyto řešení mají výhody jako větší datová komprese a snažší numerická manipulace [7]. Algoritmy naznačené v této práci jsou mimo jiné vhodné i k restauraci zvukových signálů.

1.1 Definice základních pojmů

Problematika se opírá o pojmy vysvětlené v [7]. Nejdůležitější je bezesporu tzv. „řídký vektor“ \mathbf{x} . Ten je vyjádřen jako $\mathbf{x} = [x_1, \dots, x_n]$ o velikosti N . Za pomoci slovníku \mathbf{D} a pozorovaného signálu \mathbf{y} je vyjádřen jako běžná soustava lineárních rovnic $\mathbf{D}\mathbf{x} = \mathbf{y}$. Přičemž $\mathbf{y} \in \mathbb{C}^m$ a matice $\mathbf{D} \in \mathbb{C}^{m \times N}$.

Dále je ale nutné určit další výrazy, jako 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 $\mathbf{x} = [x_1, \dots, x_5] = [2, 0, 0, 7, 0]$.

Velice důležité jsou jednotlivé normy vektoru l_p , které lze definovat jako

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

V tomto případě jsou stěžejní především normy l_0 a l_1 . Tedy počet nenulových prvků vektoru \mathbf{x} pro l_0 a součet nenulových prvků pro l_1 . Jak je patrné, vektor $\mathbf{x} = [0, 5, 3, 2, 0, 1, 0]$ bude v normě $\|\mathbf{x}\|_2 = \sqrt{(0^2 + 5^2 + 3^2 + 2^2 + 0^2 + 1^2 + 0^2)} \doteq 6,245$.

Pro určování vzdálenosti matic se často používá Frobeniova norma [6]

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}. \quad (1.3)$$

Dále je nutné určit stupeň řídkosti, do kterého bude \mathbf{x} dostatečně řídké, případně se bude pokračovat v jeho zpracování. 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

$$\|\mathbf{x}\|_0 \leq K, \quad (1.4)$$

kde K je maximální počet nenulových složek (Sparsity level). Relativní řídkostí se rozumí poměr $\frac{K}{N}$. Množina všech K -řídkých vektorů se značí jako

$$\sum_K^N := \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_0 \leq K\}. \quad (1.5)$$

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

$$\sigma_k(\mathbf{x}) := \sigma_k^N(\mathbf{x})_p := \inf_{\mathbf{z} \in \sum_k^N} \|\mathbf{x} - \mathbf{z}\|_p. \quad (1.6)$$

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 jakoukoliv normu.

1.2 Řídká řešení a podmínky

Hlavní myšlenka řešení vychází z potřeby získat co nejřidší řešení. Požadované (přípustné) řešení je tedy vektor \mathbf{x} o dostatečné řídkosti (K -řídký). Základní podmínka označovaná jako (P_0) je [4][7]

$$\min_x \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \quad (1.7)$$

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 (Atoms). Jak se slovník získá popisuje kapitola 3. Možných reprezentací \mathbf{y} je nekonečně mnoho, proto jsou zajímavé právě ty nejřidší nebo alespoň ty s drobnou odchylkou δ

$$\min_x \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_p \leq \delta. \quad (1.8)$$

Problém ale nastává v užití nějakého z algoritmů konvexní optimalizace [7]. Protože norma l_0 není konvexní funkce, nelze na ní žádný tento algoritmus aplikovat. Ovšem nejbližší norma l_1 už konvexní je,

$$\min_x \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_p \leq \delta, \quad (1.9)$$

proto se hledají podmínky, při kterých se řešení obou úloh v normách shodují. Tyto podmínky jsou naznačeny v [7].

2 REKONSTRUKCE AUDIO SIGNÁLU

Tato kapitola se zabývá rekonstrukcí zvukového 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 (CD, magnetické pásky, gramofonové desky) mohou být některé shluky dat špatně čteny a vzniknou zkreslené nebo vynechané informace. Takové chyby vznikají také při přenosu přenosovým médiiem (například telefonní linka - typicky u internetové VoIP telefonie).

Chyby v audio signálu se projeví různými způsoby. Mezi nejčastější patří šum s praskáním, přebuzené vlny a u VoIP ztráta paketů. V takových případech lze zřetelný signál vyhodnotit, tyto artefakty najít a následně rekonstruovat. Rekonstrukcí se myslí vhodnou technikou signál opravit nebo docílit co největší podobnosti s původním nepoškozeným signálem.

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

2.1 Označení poškozených vzorků

Audio signál je ovlivněný pouze v některých místech. Aby se daly opravit, je nutné tyto části detekovat. Poškozený signál definovaný vektorem $\mathbf{y} \in \mathbb{R}^L$ vychází z původního neovlivněného vektoru $\mathbf{s} \in \mathbb{R}^L$. Po navzorkování průběhu je $I \in \{1, 2, \dots, L\}$ definovaná jako množina vzorků, kde I^m jsou poškozené prvky, I^r nepoškozené prvky, přičemž $\{I^r, I^m\} \in I$. Dále vektor \mathbf{n} představuje zkreslení signálu (chybějící vzorky, přebuzení). Lze napsat

$$\begin{aligned}\mathbf{y}^r &= \mathbf{y}(I^r) = \mathbf{s}(I^r), \\ \mathbf{y}^m &= \mathbf{y}(I^m) = \mathbf{s}(I^m) + \mathbf{n}.\end{aligned}\tag{2.1}$$

Pomocí lineárního modelu s maticemi se data dají vyjádřit jako

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

Matice \mathbf{M}^r je tzv. „matice měření“. Sestaví se jako jednotková diagonální matice o rozměru $L \times L$ s hodnotou jedna v diagonále na řádce \mathbf{M}_i^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 $L = 8$, 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 6×6 tedy \mathbf{I}_6 a vzorků I^r, I^m , jsou sestaveny matice

$$\mathbf{M}^{r,6 \times 6} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.3)$$

$$\mathbf{M}^{m,6 \times 6} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.4)$$

2.2 Segmentové zpracování

Aby bylo k dispozici více nezávislých dat, použitelných pro modelování průběhu, je vhodné signál rozdělit do více segmentů. Podle [1] je přiměřená velikost zpracovávané části 64 ms s přesahem 75 %. Ovšem každý segment dat je zpracován nezávisle na ostatních. Z celkového signálu jsou jednotlivé úseky získané pomocí násobícího okénka.

Tyto rámce, o délce $N \in \mathbb{Z}$, s indexem i , získané z celkového průběhu L , jsou tvořené vzorky I^r a I^m . Z nich následně $\mathbf{y}_i \in \mathbb{R}^N$, $N \ll L$. Pomocí násobícího okna analýzy \mathbf{w}_a je z celkového průběhu získán rámeček $\mathbf{y}_i(t) = \mathbf{y}_i(t + t_i)$, $0 \leq t \leq N - 1$, respektive $\mathbf{s}_i(t) = \mathbf{s}_i(t + t_i)$, $0 \leq t \leq N - 1$. Díky rovnici (2.2) lze na takový rámeček aplikovat

$$\mathbf{y}_i^r = \mathbf{M}_i^r \mathbf{s}_i. \quad (2.5)$$

Matice měření \mathbf{M}_i^r je submatice o rozměru $N \times N$ z původní \mathbf{M}^r . Po zpracování pomocí metod naznačených v sekci 2.4 se jednotlivé segmenty opět díky OLA (Overlap-add) vhodně poskládají [9]. Hledaný signál \mathbf{s}_i je pomocí těchto naznačených algoritmů aproximován jako $\hat{\mathbf{s}}_i$. Jednotlivé rámce jsou pomocí násobícího okénka syntézy \mathbf{w}_s sestavené podle

$$\hat{\mathbf{s}}_i = \frac{\sum_i \mathbf{w}_s(t - t_i) \hat{\mathbf{s}}_i(t - t_i)}{\sum_i \mathbf{w}_s(t - t_i)}. \quad (2.6)$$

Použité druhy okénkových funkcí:

Pravoúhlé násobící okénko (Rectangular) je nejjednodušší okénková funkce. Jeho výhoda je především to, že nijak nemodifikuje signál se kterým se násobí. Proto je vhodné ho použít jako okno analýzy \mathbf{w}_a . Obecný vztah

$$\mathbf{w}(I) = 1. \quad (2.7)$$

Sinusové násobící okénko (Sine) je použitý pro okno syntézy \mathbf{w}_s . Jeho definice

$$\mathbf{w}(I) = \sin\left(\frac{\pi I}{N-1}\right). \quad (2.8)$$

2.3 Řídké reprezentace pro rekonstrukci zvukových segmentů

Podle [7] je původní signálový vektor vyjádřen lineární kombinací řídkého sloupcového vektoru \mathbf{x}_i a slovníku \mathbf{D} . Tedy je možné i signálový segment definovat jako

$$\mathbf{s}_i \approx \mathbf{D}\mathbf{x}_i. \quad (2.9)$$

Vektor řídkých řešení \mathbf{x}_i je nejjednodušší hledané řešení i -tého rámce. Po kombinaci s rovnicí (2.5) je možné získat vztah, který řeší řídký model pro segment obsahující neovlivněné vzorky

$$\mathbf{y}_i^r = \mathbf{M}_i^r \mathbf{s}_i \approx \mathbf{M}_i^r \mathbf{D}\mathbf{x}_i. \quad (2.10)$$

Zjištěním řídkého vektoru \mathbf{x}_i díky I^r a rámcové matice měření \mathbf{M}^r je vlastně možné odhadovat a modelovat podobu původního signálu \mathbf{s}_i . Tento získaný průběh $\hat{\mathbf{s}}_i$ je tedy zjištěn ze vztahu

$$\hat{\mathbf{s}}_i(I_i^m) \approx \mathbf{M}_i^m \mathbf{D}\hat{\mathbf{x}}_i. \quad (2.11)$$

Tyto vztahy jsou známé z [1].

2.4 Aproximační algoritmy

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

$$\hat{\mathbf{x}}_i = \arg_{\mathbf{x}_i} \min \|\mathbf{x}_i\|_0 \text{ na základě } \|\mathbf{y}_i^r - \mathbf{M}_i^r \mathbf{D}\mathbf{x}_i\|_2^2 \leq \theta_i^\epsilon. \quad (2.12)$$

Problém je ale v řešení tohoto vektoru. Přestože je vztah přímo neřešitelný, může být nahrazen aproximačním algoritmem. Známých a použitelných metod je v současné době více [7]. 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][5]:

- 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 2.4.1. Metody MP jsou zjednodušené OMP, Weak-OMP zase zjednodušené MP. Tyto varianty jsou jednodušší na výpočet ale přinášejí horší výsledky. Protože v této práci není kladen důraz na rychlost ale spíše na účinnost a výkonnost, bude se dále řešit OMP.

2.4.1 Algoritmus OMP

Metoda, řešící řídký vektor \mathbf{x}_i , OMP je založena na aproximaci, kdy 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 $\mathbf{y}_i^r = \mathbf{M}_i^r \mathbf{D} \mathbf{x}_i$ [1] [2] [5]. V tabulce 2.1 je znázorněný postup algoritmu OMP. Nejprve jsou programu dodány vstupní parametry: vektor nepoškozeného průběhu, matice měření pro vektor \mathbf{y}_i^r , maximální počet nenulových složek \mathbf{x} viz (1.4) a chyba nejlepší aproximace viz (1.6).

Prvním krokem je určení počátečních podmínek. Slovník $\tilde{\mathbf{D}}$ vznikne vynásobením \mathbf{M}_i^r , původní podobou slovníku \mathbf{D} a maticí \mathbf{W} . Diagonální matice \mathbf{W} je složena z prvků $\mathbf{W}_{jj} = \|\mathbf{M}_i^r \mathbf{d}_j\|_2^{-1}$ 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 θ_i^ϵ . Jinými slovy bude buď dosaženo vektoru o dostatečné řídkosti nebo chyby o přijatelné velikosti.

Tab. 2.1: Algoritmus OMP

| |
|---|
| <p>Vstupní parametry: $\mathbf{y}_i^r, \mathbf{M}_i^r, \mathbf{D} = \{\tilde{\mathbf{d}}_j\}_{j \in \Gamma}, K_{\max}, \theta_i^\epsilon$</p> |
| <p>První inicializace: Slovník: $\tilde{\mathbf{D}} = \mathbf{M}_i^r \mathbf{D} \mathbf{W}$. Čítač iterací $k = 0$. Nosič vektoru $\Omega_0 = \text{supp}(\mathbf{x}_0) = \emptyset$. Rezidua $\mathbf{r}_0 = \mathbf{y}_i^r$.</p> |
| <p>Hlavní cyklus: Dvě podmínky $k = K_{\max}$ nebo $\ \mathbf{r}_k\ _2^2 < \theta_i^\epsilon$. Inkrementace čítače iterací $k = k + 1$. Výběr atomu: hledej $j = \arg \max_j \langle \mathbf{r}_k, \tilde{\mathbf{d}}_j \rangle$. Obnovení nosiče vektoru $\Omega_k = \Omega_{k-1} \cup j$. Nalezení řešení v kroku k, $\mathbf{x}_k = \arg \min_{\mathbf{u}} \ \mathbf{y}_i^r - \tilde{\mathbf{D}}_{\Omega_k} \mathbf{u}\ _2$. Aktualizace reziduí $\mathbf{r}_k = \mathbf{y}_i^r - \tilde{\mathbf{D}}_{\Omega_k} \mathbf{x}_k$.</p> |
| <p>Výstupní parametry: $\mathbf{x}_i = \mathbf{W} \mathbf{x}_k$</p> |

2.5 Výpočet SNR

K vyhodnocení výkonnosti těchto různých algoritmů může být použito SNR (odstup signálu od šumu – Signal-to-Noise Ratio). V našem případě SNR reprezentuje spíše „odstup opraveného od poškozeného signálu“ [1]. Vypočítá se z hodnot vektoru poškozeného $\mathbf{s}(I^c)$ a rekonstruovaného signálu $\hat{\mathbf{s}}$ jako

$$\text{SNR}_c(\mathbf{s}, \hat{\mathbf{s}}) = 10 \log \frac{\|\mathbf{s}(I^c)\|_2^2}{\|\mathbf{s}(I^c) - \hat{\mathbf{s}}(I^c)\|_2^2}. \quad (2.13)$$

2.6 Výpočet RMSE

V dalších částech bude zmínka o výpočtu RMSE (odmocnina ze střední kvadratické chyby). Jedná se o často používanou hodnotu pro určení kvality predikce signálu (mezi referenčním a odhadovaným) [8]. Těmto rozdílům se také říká rezidua.

Referenční signál je vektor \mathbf{x} a odhadovaný $\hat{\mathbf{x}}$

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

$$\text{RMSE}(\hat{\mathbf{x}}, \mathbf{x}) = \sqrt{\text{E}((\hat{\mathbf{x}} - \mathbf{x})^2)} = \sqrt{\frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{n}}. \quad (2.14)$$

3 SLOVNÍKY

Pro nalezení nejlepší aproximace signálu \mathbf{x} je nutné použít vhodný slovník \mathbf{D} . Tato matice je soubor předem známých hodnot (diskrétních frekvencí). Volba slovníku je podstatná část při snaze najít co možná nejjednodušší \mathbf{x} pro reprezentaci libovolného signálu. K vytvoření \mathbf{D} lze přistupovat dvěma různými způsoby.

- Statické slovníky (DCT, Gaborův)
- Přizpůsobené slovníky (K-SVD, ILS-DLA, RLS-DLA, MDCT, ODL, MOD)

Oba přístupy mají své výhody a nevýhody. V bakalářské práci jsou tyto metody rozebrány a dále prakticky vyzkoušeny na rekonstrukce zvukových signálů.

3.1 Statické slovníky

Statické slovníky jsou matice diskrétních koeficientů sestavených z pevné funkce. Podle zvolené funkce v sobě jednotlivé koeficienty nesou potřebné informace, jako jsou například frekvence a fáze. S větším množstvím těchto informací teoreticky rostou i možnosti a přesnost rekonstrukce libovolného signálu.

3.1.1 DCT slovník

První ze statických slovníků založených na řešení řídkých vektorů je DCT slovník (Discrete Cosine Transform). Vyniká především svou jednoduchostí a nenáročnou aplikací. Jeho matice je složena ze sloupcových vektorů podle funkce $\mathbf{D}^c = [\mathbf{d}_0^c, \dots, \mathbf{d}_{K_c-1}^c]$. Tyto sloupce j jsou z rozsahu $0 \leq j \leq K_c - 1$. Pro každý sloupec o délce N platí $0 \leq t \leq N - 1$ jako

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

Parametr K_c značí počet sloupců matice. Prakticky tedy slovník udává poměrně malý počet diskrétních 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 velice často používán v oblasti komprese zvukových a obrazových signálů. Jeho aplikací je dosaženo kompaktní spektrální reprezentace [8][2].

3.1.2 Gaborův slovník

Oproti DCT slovníku, Gaborův nese ve své definici navíc informace o fázi každé frekvence. Jeho vytvoření je podle $\mathbf{D}^g = \{\mathbf{d}_{(j,\varphi)}^g\}_{(j,\varphi) \in \Gamma}$ pro sadu parametrů $\Gamma = [0, K_g - 1] \times [0, 2\pi]$, kde K_g určuje rozměr slovníku. Jedná se tedy o matici, která má další rozměr pro fázi kmitočtu φ . Složky matice se zjistí z

$$\mathbf{d}_{j,\varphi}^g(t) = \mathbf{w}_d(t) \cos\left(\frac{\pi}{K_g} \left(t + \frac{1}{2}\right) \left(j + \frac{1}{2}\right) + \varphi\right). \quad (3.2)$$

Jak je patrné v [2], užití Gaborova slovníku přináší často řidší, tedy přijatelnější a lepší výsledky. Aplikace v algoritmu je ale složitější. Kvůli fázi musí být rozšířen i řešený vektor a proto je nasazení tohoto slovníku komplikovanější. Navíc se výrazně zvyšuje náročnost na výpočet.

3.2 Přizpůsobené slovníky

Jedná se o slovníky vytvořené na míru nějakému signálu. Prakticky se používají algoritmy, které vytvoří libovolný slovník a následně jej natrénují sadou dostupných dat. Teoreticky pak vznikne velice 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 práce se zaměří na užití slovníku pro zvukové signály.

3.2.1 Sjednocené MDCT báze

Reprezentace signálu je v tomto případě řešena vícebázovým DCT slovníkem [11]. Struktura \mathbf{D} je složená z M bloků slovníku

$$\mathbf{D} = \bigcup_{m=0}^{M-1} \mathbf{D}_m. \quad (3.3)$$

Každý blok MDCT se skládá z atomů podle

$$\mathbf{D}_m = \{\mathbf{g}_{m,p,k} | 0 \leq p < P_m, 0 \leq k < L_m\}, \quad (3.4)$$

kde m představuje index bloku, p index segmentu a k index frekvence.

Bloky MDCT jsou v podstatě totožné se statickým DCT slovníkem. Liší se ovšem ve škálovatelnosti šířky okénkové funkce $\mathbf{w}(u)$. Délka m -tého bloku je polovina okna analýzy

$$L_m = L_0 2^m, \quad (3.5)$$

a každý atom bloku

$$\mathbf{g}_{m,p,k}(n) = \mathbf{w}_m(u) \sqrt{\frac{2}{L_m}} \cos \left[\frac{\pi}{L_m} \left(u + \frac{1 + L_m}{2} \right) \left(k + \frac{1}{2} \right) \right], \quad (3.6)$$

s

$$u = n - pL_m - \frac{L_m}{2}. \quad (3.7)$$

Myšlenka použití vícebázového slovníku spočívá v rozdělení zpracovávaného segmentu signálu na různě dlouhé části podle průběhu. Každá část je při reprezentaci aproximována jiným blokem MDCT. Jak je naznačeno výše, různé bloky používají jiné délky okénkové funkce (s následujícím blokem je šířka poloviční).

Tímto přístupem lze při analýze signálu s rozdělením na proměnné úseky dlouhé statické části aproximovat pomocí bloku s dlouhými okénky a naopak krátké proměnné úseky aproximovat blokem s krátkými okénky.

Podle [11] je optimální počet bází $M = 8$ pro délky okének mezi 2,9 až 370 ms. S větším počtem bloků roste složitost a zároveň nedochází k výrazně lepším aproximacím.

3.2.2 K-SVD

Metoda K-SVD pro přizpůsobení slovníku vychází z principu zobecněného algoritmu K-means pro kvantizaci vektorů a SVD 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 zafixuje \mathbf{x} a hledá se vhodná kombinace atomů [3]. Detailnímu popisu algoritmu se věnuje kapitola 4.

3.2.3 RLS-DLA

Další z řady algoritmů využívá rekurzivní metodu nejmenších čtverců – Recursive Least Squares Dictionary Learning Algorithm (RLS-DLA). Ta je v první části totožná s algoritmem K-SVD, ve druhé ale namísto SVD rozkladu signálu používá metodu nejmenších čtverců. Pro hledání $f(\mathbf{D}) = \|\mathbf{Y} - \mathbf{DX}\|_F^2$ při pevné sadě řídkých vektorů \mathbf{X} je řešení

$$\mathbf{D} = (\mathbf{YX}^T)(\mathbf{XX}^T)^{-1}. \quad (3.8)$$

Metoda pro trénování slovníku RLS-DLA vychází ze starší Iterative Least Squares Dictionary Learning Algorithm (ILS-DLA). Ovšem hlavní rozdíl oproti ILS-DLA a K-SVD je v použití jiného přístupu. Zatímco se dříve zmíněné slovníky nejprve sestaví a natrénují na sadu signálů, RLS-DLA se obměňuje postupně. Tedy s každým zpracovávaným vektorem se atomy slovníku sestavují znovu. Proto je algoritmus označován jako „rekurzivní“ [12][13].

3.2.4 ODL

Odlišným přístupem vyniká Online Dictionary Learning (ODL) [10]. Předchozí slovníky využívají konečný počet vzorků signálu k nalezení optimálního slovníku. To ale v případě požadavku na velký rozměr slovníku a tedy i velkého množství trénovacích dat může být problém. Velké problémy nastanou při potřebě dynamického trénování \mathbf{D} pro více signálů. V takovém případě je nutné generovat a trénovat na každý signál nový slovník a to je pro tisíce signálů nepraktické. Typicky například pro video sekvenci nebo spoustu krátkých zvukových signálů.

Slovník ODL zde funguje podobně jako RLS-DLA ale pro reprezentaci kratších signálů se vybírá a trénuje pouze jeden atom (případně menší submatice z \mathbf{D}). Takový přístup oproti ostatním metodám přináší rychlejší výpočet ale za cenu horší aproximace. Nicméně k některým aplikacím, kde je vyžadována rychlost může být vhodná.

4 NATRÉNOVÁNÍ K-SVD

Úkolem práce je zhodnotit a srovnat různé druhy slovníku pro řídké reprezentace k rekonstrukci zvukového signálu. Jako zástupce přizpůsobených slovníků byl zvolen K-SVD. Protože po natrénování \mathbf{D} vzniká strukturou podobná matice jako DCT, je jeho zapojení do rekonstrukčního algoritmu OMP jednodušší než například MDCT.

Hlavním důvodem zvolení K-SVD je volná dostupnost toolboxu k učení slovníku (viz kapitola 5.2). Díky sadě funkcí realizující samotný algoritmus je možné se zaměřit spíše na nastavování vhodných parametrů a samotné srovnání slovníků [3].

4.1 Kvantizace vektorů a K-means

Princip K-SVD algoritmu řeší problematiku kvantizace vektoru VQ (vector quantization nebo také „clustering“). Kvantizace vektorů je vlastně 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 může být podle matice reziduí \mathbf{E} , kde

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

Rezidua zde představují číselnou chybu reprezentace signálu pomocí slovníku \mathbf{D} a matice řídkých vektorů \mathbf{X} . S lépe reprezentujícími \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} [3].

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} poměrně obtížný 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 $\mathbf{D}\mathbf{X}$ podle

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

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 $\mathbf{D}\mathbf{X}$ 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 měření $\boldsymbol{\Omega}_k$ 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 .

4.2.2 Použití SVD

Nyní se práce zaměří na vhodnou úpravu atomů \mathbf{d}_k . Momentálně je dosaženo stavu, kdy se dá minimalizace 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 .

Metoda použitá k rozkladu matice je Singulární rozklad (SVD). Po aplikaci dojde k rozebrání \mathbf{E}_k^R do tří matic

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

kde $\mathbf{U} \in \mathbb{R}^{n \times n}$ a $\mathbf{V} \in \mathbb{R}^{|\omega_k| \times |\omega_k|}$ jsou unitární matice a $\boldsymbol{\Sigma}$ je diagonální matice s kladnými reálnými prvky. Podrobný popis singulárního rozkladu matic lze nalézt v [6].

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 $\boldsymbol{\Sigma}(1, 1)$

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

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

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

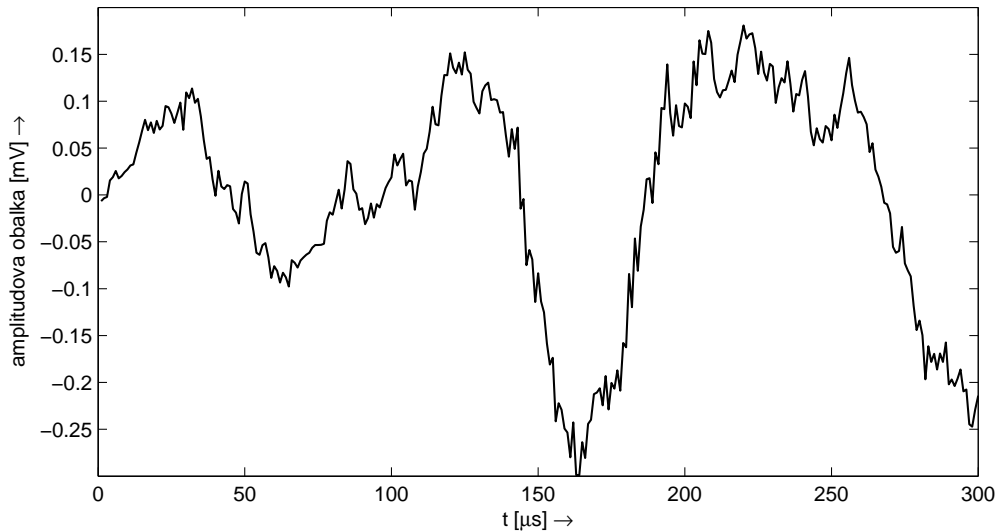
Tab. 4.1: Shrnutí K-SVD algoritmu

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

4.3 Sada vzorků

Důležitým prvkem při optimalizaci slovníku je vhodná volba sady trénovacích dat \mathbf{Y} . Struktura odpovídá matici složené z N vzorků. Délka vzorků odpovídá šířce slovníku (počtu atomů) a počet vzorků musí být větší než počet atomů.

Mějme původní zvukový signál bez jakýchkoliv vnějších zkreslení. Takový náhodný signál může vypadat například jako na obrázku 4.1. V ideálním případě by



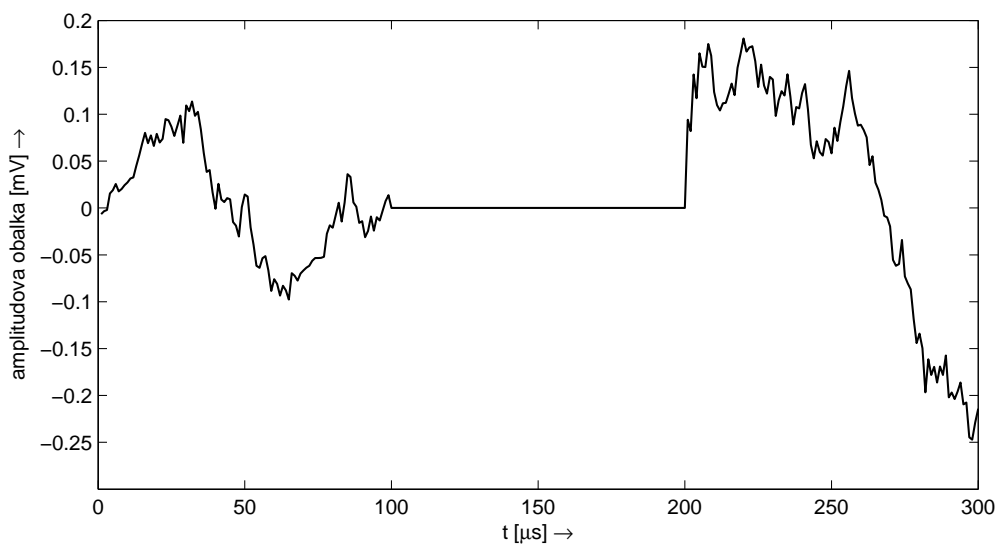
Obr. 4.1: Původní signál

bylo nejlepší sadu vzorků pro natrénování slovníku získat z tohoto průběhu. To je ale možné pouze v laboratorních podmínkách, kdy je původní nepoškozený signál k dispozici.

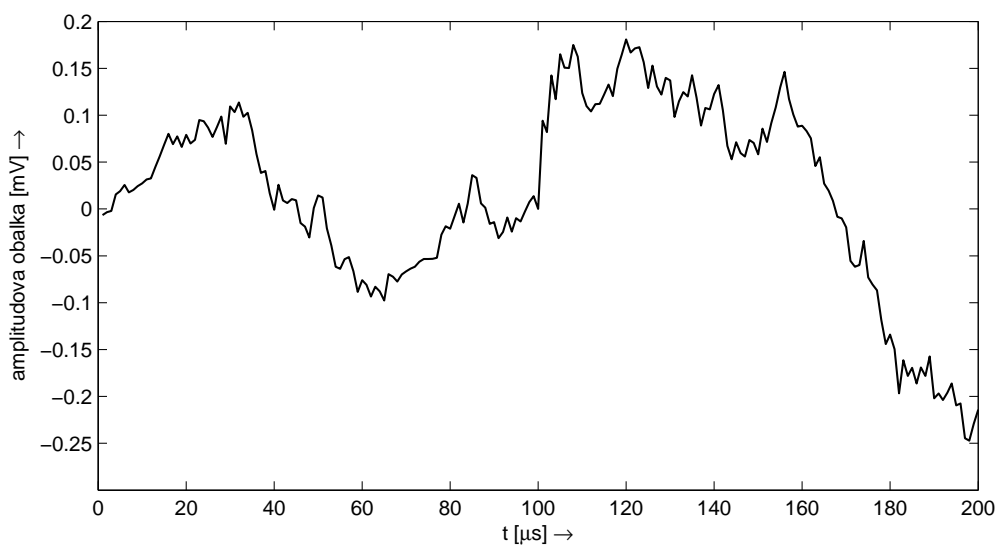
Reálně ovšem bývá dostupný pouze poškozený signál, protože se snažíme o jeho rekonstrukci. Příklad takového signálu je na obrázku 4.2. Zde jsou vidět již chybějící vzorky mezi 100 až 200 μs , reprezentované nulou. Sestavit segmenty z poškozeného průběhu není vhodné. V přítomnosti chybných vzorků nemá smysl slovník na tyto zkreslení optimalizovat.

Jednoduchým řešením je před sestavováním sady vzorků všechny chybné údaje úplně vypustit. V laboratorních podmínkách se ovlivněné části odstraní jednoduše. Obvykle na signál ukazuje maska zkreslených vzorků. Podle masky je možné přesně identifikovat chyby a ze signálu je jednoduše smazat. Pro rekonstrukce zvuků u kterých nejsou známé původní průběhy (ve skutečnosti všechny) musí být tato maska vytvořena.

Signál po odstranění chybných vzorků lze vidět na obrázku 4.3. Je zřejmé, že došlo také ke zmenšení délky z 300 na 200 μs .



Obr. 4.2: Poškozený signál



Obr. 4.3: Trénovací signál

4.3.1 Sestavení vzorků

Matice \mathbf{Y} obsahující vzorky signálu má tyto úseky uloženy po sloupcích. K sestavení sady vzorků může být signál rovnoměrně rozdělen (například po 256 vzorcích) do segmentů. Ovšem pro krátké zvuky (řádově několik sekund) nebude počet segmentů dostatečný.

Příklad: máme zvukový signál o délce 1 s o vzorkovací frekvenci $f_{vz} = 8$ kHz. Tedy na jednu vteřinu připadá 8 000 vzorků signálu. Při rovnoměrném rozdělení na segmenty o délce 256 vzorků budeme schopni sestavit pouze 31 celých segmentů. To s požadavkem počtu segmentů vzorků větším než počet atomů slovníku $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$ ($N \gg K$), kdy slovník pro takovou aplikaci bude mít 256 atomů nebude stačit.

Řešením je použití posuvu o méně vzorků než celý segment tzv. „shift“. Při posuvu $shift = 16$ vzorků je již možné vytvořit dostatek segmentů do matice \mathbf{Y} . Pro výše zmíněné parametry je proto ideální mít signály o délce alespoň 10 s, aby nebylo nutné posuv realizovat.

5 IMPLEMENTACE V PROSTŘEDÍ MATLAB

Práce se zvukovými signály a jejich rekonstrukce pomocí popsanych metod je možná díky sadě funkcí nazvaných Audio Inpainting Toolbox. Tento toolbox je sestaven ze zdrojových kódů tzv. M-filů, kdy každý soubor představuje dílčí program řešící danou problematiku. Jedná se o sadu volně použitelných nástrojů s licenci GNU Public Licence verze 3, určených především pro demonstraci rekonstrukce zvukových signálů na základě řídkých reprezentací.

Vytvoření a natrénování K-SVD slovníku zajišťuje ksvd toolbox. Oba balíčky jsou již přítomny v kompletním SMALLboxu, volně dostupném na adrese autorů <http://small-project.eu/>.

Verze použitých součástí:

- SMALLbox v. 1.9
- Audio Inpainting Toolbox v. 1.0
- ksvdbox v. 13
- ompbox v. 10.

5.1 Audio Inpainting Toolbox

K práci s toolboxem je nutné mít nainstalovaný Matlab alespoň ve verzi R2010a. Na nižších není zaručena funkčnost. Funkce s konkrétními řešícími algoritmy navíc vyžadují nainstalovaný CVX toolbox přítomný ve SMALLboxu. Soubory jsou rozděleny do složek podle oblastí svých použití. Umístění AIT v adresářové struktuře je „/toolboxes/AudioInpaintingToolbox/“.

5.1.1 Metody řešení

Jednotlivé algoritmy pro hledání řídkého vektoru jsou ve složce „Solvers/“. Zavoláním jména v příkazovém okně matlabu dojde k jejich spuštění. V každé funkci jsou na začátku výpisy vstupních a výstupních parametrů, se kterými se pracuje.

Například `inpaintFrame_OMP.m` i `inpaintFrame_OMP_Gabor.m` pracují s parametry „problemData“, který určuje vstupní signál a „param“ pro nastavení slovníku a okénkové funkce. Zadané „problemData“ v sobě nese samotný řešený signál a masku označující chybné vzorky. Všechny přítomné metody pracují pouze se segmenty s výjimkou `inpaintSignal_IndependentProcessingOffFrames.m`. Ta zpracovává celý signál a realizuje rozdělení na segmenty, volání OMP a následné skládání za pomoci OLA.

Ve složce „Utils/“ se nachází pomocné funkce `wRect.m` a `wSine.m`. Podložka „Dictionaries/“ obsahuje knihovny pro výpočet DCT a Gaborova slovníku.

5.1.2 Experimenty

Příklady experimentů jsou obsaženy ve složce „Experiments/“. Všechny pracují s jednokanálovou zvukovou vlnou uloženou ve formátu .wav. První z nich ve složce „DeclippingExperiment/“ řeší přebuzený signál. Doplněním chybějících vzorků se zabývá `MissingSampleTopologyExperiment.m` v dalším adresáři. Ten ovšem z celkového signálu vybírá náhodné segmenty a na nich demonstruje rekonstrukční metody. Kvůli náhodné volbě segmentů v každém měření ale není vhodný ke statistickému vyhodnocení.

Proto byl původní toolbox doplněn o experiment `InpaintingExperiment.m`. Při zavolání jména experimentu lze určit několik klíčových parametrů zobrazených v tabulce 5.1.

Tab. 5.1: Parametry `InpaintingExperiment`

| Vstupní parametr | Popis |
|---|--|
| <code>expParam.N</code> | velikost zpracovávaných segmentů |
| <code>expParam.totalMissSamplesList</code> | šířky děr k testování |
| <code>expParam.position</code> | pozice začátku generování díry |
| <code>expParam.soundDir</code> | cesta k souborům |
| <code>expParam.solvers</code> | struktura metod a parametrů k rekonstrukci |
| Volání: <code>InpaintingExperiment(expParam)</code> | |

5.2 K-SVD toolbox

Implementace K-SVD algoritmu k natrénování slovníku pro řídké reprezentace je možná díky `ksvdboxu`. Toolbox je umístěn v „/toolboxes/KSVD/ksvdbox“. Některé funkce `ksvdboxu` nutné pro spuštění trénování slovníku vyžadují MEX kompilátor. Ve složce „private/“ se nachází soubory, které musí být zkompileovány. Při správně nainstalovaném kompilátoru (například Microsoft Visual C++) stačí spustit `start_ksvd.m` v adresáři `SMALLboxu` a soubory se zkompilují. Dále je nutné pro hledání reprezentačního vektoru algoritmus OMP, který je vytvořen v toolboxu „/toolboxes/KSVD/ompbox“.

5.2.1 Práce s toolboxem

Proces učení slovníku realizuje funkce `ksvd.m`. K lepší manipulaci a jednoduššímu volání byla vytvořena další funkce `KSVD_Dictionary.m`. Ta analogicky jako funkce

DCT_Dictionary.m a Gabor_Dictionary.m vrací slovník vhodný pro rekonstrukci chybějících vzorků signálu. Parametry k natrénování slovníku jsou v tabulce 5.2.

Tab. 5.2: Parametry K-SVD

| Vstupní parametr | Popis |
|---|--|
| trainData | vektor s daty pro natrénování slovníku |
| shift | posuv o |
| param.KSVDparam.initdict | počáteční slovník |
| param.KSVDparam.Tdata | řídkost dosahovaných vektorů |
| param.KSVDparam.iternum | počet iterací |
| param.N | šířka zpracovávaného segmentu |
| param.redundancyFactor | faktor nadbytečnosti slovníku |
| Volání: $D = \text{KSVD_Dictionary}(\text{param}, \text{trainData}, \text{shift})$ | |

6 PARAMETRY NASTAVENÍ K-SVD

Následující kapitola se zabývá vhodnou volbou parametrů k natrénování K-SVD slovníku. Snaha je především o určení kolik iterací, jakou řídkost, počáteční slovník a přesah segmentů použít k optimální funkci.

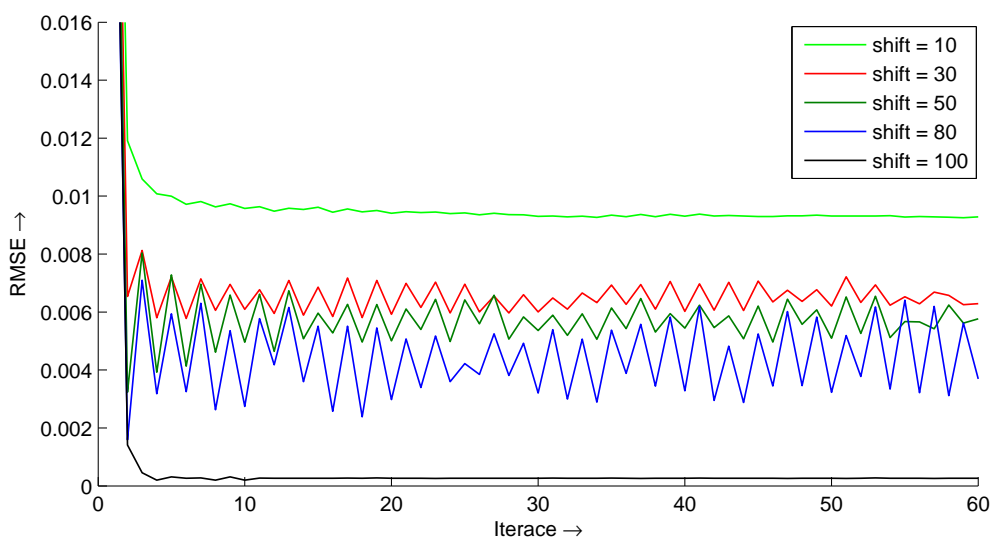
Pokusy budou realizovány pro zvukový signál `music07_16kHz.wav`. Soubor představuje jednonábovovou vlnu, ve které je zaznamenaný hudební signál s $f_{vz} = 16$ kHz.

6.1 Volba posuvu segmentů

Jak je naznačeno v kapitole 4.3.1, při zpracování kratších signálů je nutné použít posuv menší než je šířka segmentů. Nabízí se otázka, zda je lepší s menším posuvem získat velké množství trénovacích dat nebo nejvyšším možným získat méně dat. Nutno podotknout, že se jedná o umělé zvětšování množství dat a vzorky mezi jednotlivými segmenty se z části opakují.

Testovaný signál má délku $t = 5$ s a vzorkovací frekvenci $f_{vz} = 16$ kHz. K dispozici je tedy 80 000 vzorků. Pro uvažovanou šířku segmentů 256 vzorků s faktorem nadbytečnosti slovníku $redundancy = 3$ (\mathbf{D} o rozměru 256×768) můžeme posun realizovat pro hodnotu v intervalu $shift = (1; 100)$.

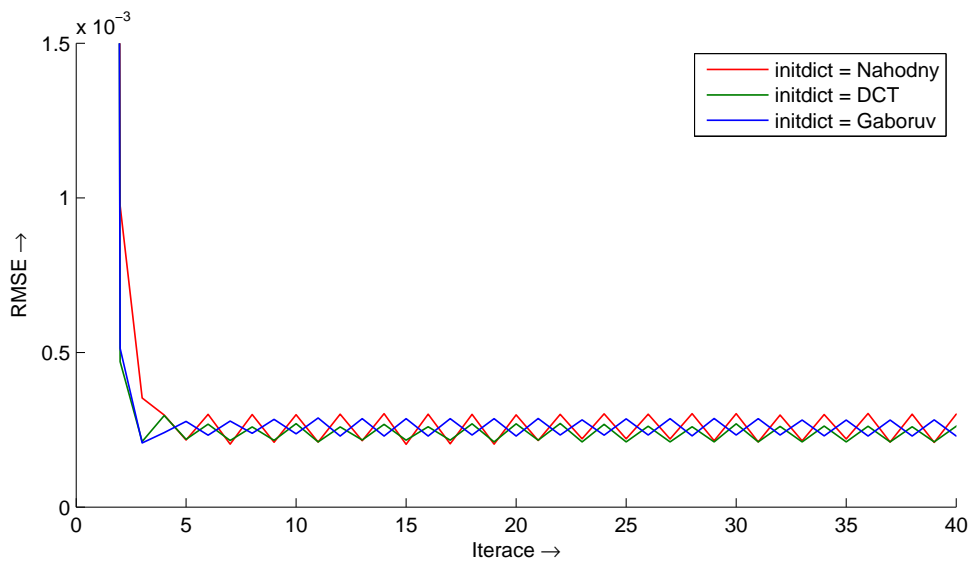
Na obrázku 6.1 je vidět účinnost trénování pro různé parametry $shift$. Z grafu lze vyvodit, že pro snížení RMSE je dobré nastavit posun na nejvyšší možnou hodnotu. Ke srovnání slovníků na signálech s podobnou délkou bude používán $shift = 100$.



Obr. 6.1: Srovnání posuvů

6.2 Volba inicializačního slovníku

Funkce algoritmu vyžaduje nějaký prvotní slovník, na který se bude K-SVD aplikovat. Zde můžou být zvoleny různé druhy. Například je možné nejprve vytvořit některý ze statických a následně jej natrénovat. Stejně tak ale lze zvolit náhodná čísla, pouze je normalizovat a nechat natrénovat. Obrázek 6.2 ukazuje účinnost natrénování K-SVD pro tři různé inicializační slovníky. Je možné vidět, že po několika prvních iteracích dochází k vyrovnání účinnosti natrénování. Dá se tedy očekávat minimální a prakticky zanedbatelný vliv volby inicializačního slovníku. Do parametru *initdict* bude v dalších pokusech generována matice složená z náhodných, normalizovaných čísel.

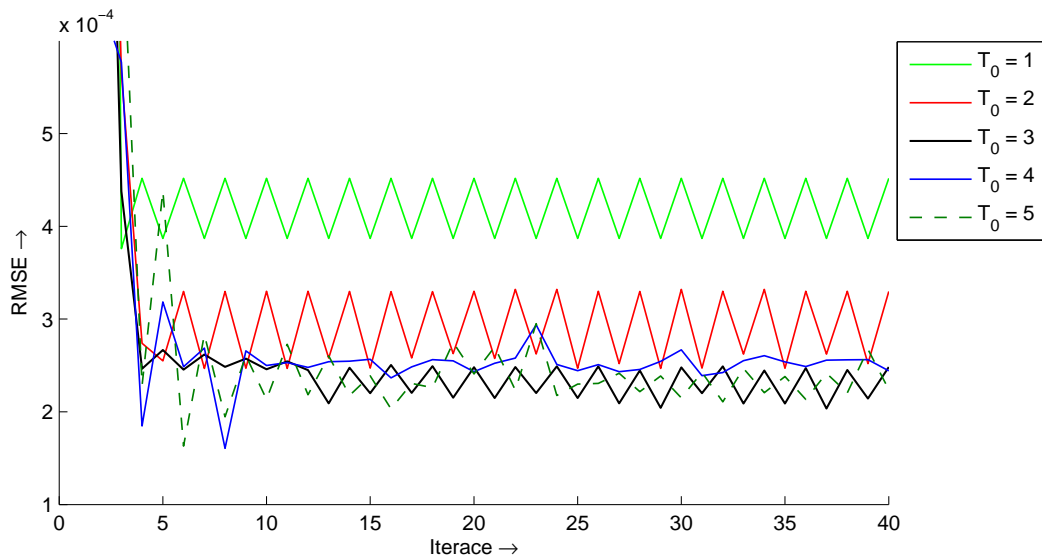


Obr. 6.2: Srovnání inicializačních slovníků

6.3 Volba řídkosti

Zvýšená pozornost by měla být věnována také parametru pro určení řídkosti T_0 z rovnice 4.3. Teoreticky, s nižší řídkostí by měl být slovník účinnější. Ovšem z důvodu popsaném v [3], nebude $T_0 = 1$ nebo $T_0 = 2$ vykazovat nejlepší výsledky.

V detailu průběhů na obrázku 6.3 jsou srovnány rozdíly slovníků trénované na stejný signál. Jednotlivé průběhy se liší pouze v nastavené řídkosti T_0 . Na obrázku je především vidět vyšší RMSE pro $T_0 = 1$ a $T_0 = 2$. Protože další nejmenší parametr $T_0 = 3$ vykazuje nižší RMSE budeme v další práci uvažovat právě takové nastavení.



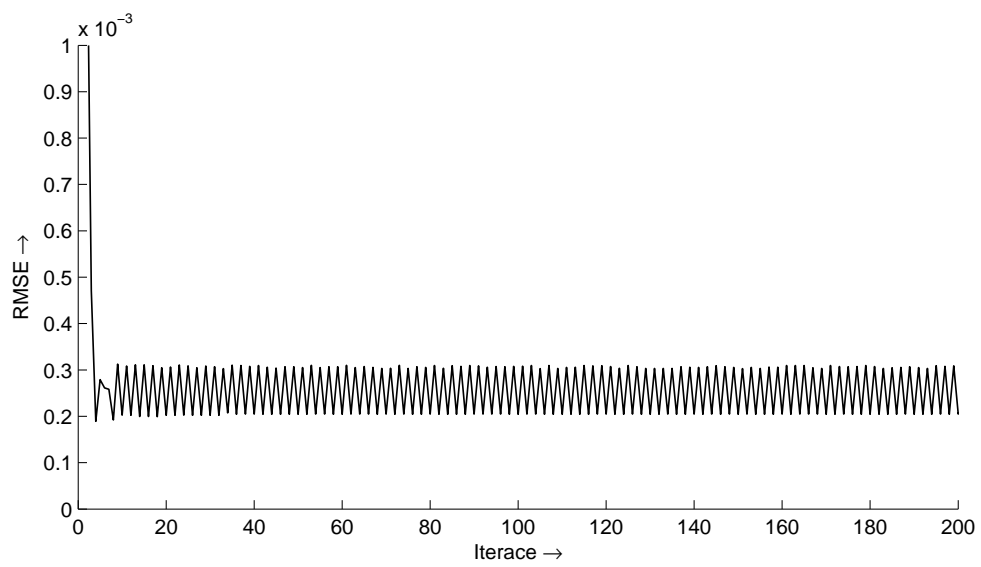
Obr. 6.3: Srovnání řídkosti

6.4 Volba počtu iterací

Algoritmus K-SVD je iterační. Tedy s každou další iterací by mělo docházet ke snižování RMSE. Hodnoty se samozřejmě po několika krocích začínají ustalovat. Proto zbytečně velké množství iterací natrénování slovníku nijak výrazně nepomůže. Je tedy nutné rozhodnout, kolik jich pro natrénování \mathbf{D} nastavit.

Na zvolený signál byl proto spuštěn K-SVD v 200 krocích. Jak lze vidět na obrázku 6.4 k ustálení dochází již při prvních iteracích. Pojem ustálení je nutné brát s rezervou, v charakteristice průběh spíše osciluje.

Pro různé signály ale může být ustálení pomalejší. Z toho důvodu je lepší nechat proběhnout více iterací jako rezervu. Podle pokusů zaznamenaných v [3] je doporučený počet 80. U dalších měření bude dále uvažován parametr $iternum = 80$.



Obr. 6.4: Průběh pro 200 iterací

7 SROVNÁNÍ SLOVNÍKŮ

Nyní se práce zaměří na rekonstrukce chybějících úseků zvukového signálu a budou srovnány výsledky za použití různých slovníků. Ze statických slovníků se porovnávají DCT a Gaborův, jako zástupce přizpůsobeného K-SVD slovník.

Experimentem budou vyhodnocovány zvukové soubory formátu .wav, v adresářích „testSpeech16kHz/“ a „testMusic16kHz/“. Jedná se o sadu řečových a hudebních signálů. Výhodou je, že všechny signály v těchto adresářích jsou stejně dlouhé (mají stejný počet vzorků – 80 000). Vzorkovací frekvence $f_{vz} = 16$ kHz. Pro měření na ně tedy budou aplikována stejná nastavení parametrů a bude pozorováno, jaké podávají jednotlivé slovníky výsledky.

7.1 Průběh experimentu

Ke srovnání slovníků byla vytvořena funkce `InpaintingExperiment.m`. Tento pokus demonstruje rekonstrukce chybějících úseků zvukového signálu a následně výsledky vyhodnocení zobrazí do grafu. Fáze experimentu probíhají v následujícím sledu:

1. inicializace parametrů,
2. načtení zvukového souboru,
3. vynulování skupiny vzorků na určeném místě,
4. postupná rekonstrukce pro všechny slovníky,
5. opakování od kroku 3 pro větší skupinu vzorků,
6. vyhodnocení a srovnání slovníků.

Parametry nastavené v experimentu lze nalézt v tabulce 7.1. Budou se zpracovávat segmenty o délce 256 vzorků, generované díry budou v rozmezí (1; 240) vzorků. Nadbytečnost slovníku je 3 (generované slovníky budou o velikosti 256×768). Okénka \mathbf{w}_d a \mathbf{w}_a jsou pravoúhlá, pouze okénko syntézy \mathbf{w}_s je sinusové.

Tab. 7.1: Nastavení experimentu

| Vstupní parametry a jejich hodnoty |
|--|
| <code>expParam.N = 256;</code> |
| <code>expParam.totalMissSamplesList = [1, 5, 10 : 10 : 240]</code> |
| <code>expParam.position = 6 000 a 33 000</code> |
| <code>param.redundancyFactor = 3</code> |
| <code>param.wd = @wRect</code> |
| <code>param.wa = @wRect</code> |
| <code>param.OLA_ws = @wSine</code> |

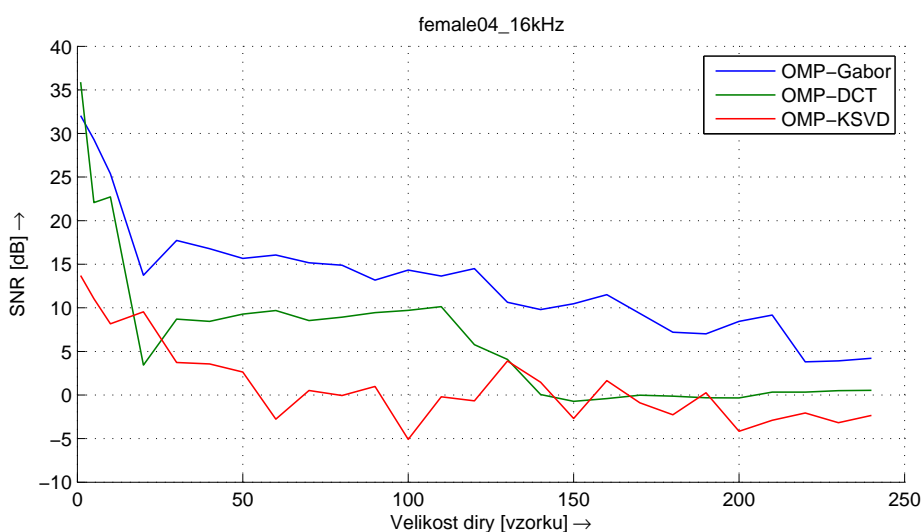
Po načtení signálu a vygenerování chyb dochází k samotné rekonstrukci. Slovníky DCT a K-SVD rekonstruují segmenty funkcemi `inpaintFrame_OMP.m` a Gaborův `inpaintFrame_OMP_Gabor.m`.

Vyhodnocení kvality rekonstrukce je prováděno srovnáním odstupů signálů SNR (viz kapitola 2.5) v závislosti na velikosti chybějícího úseku.

7.2 Výsledky srovnání

Porovnání bude provedeno pro řečové signály v souborech `female04_16kHz.wav` a `male04_16kHz.wav`. První představuje ženský, druhý mužský anglicky mluvený hlas. Generované díry se začnou vytvářet od pozice 6 000. vzorku.

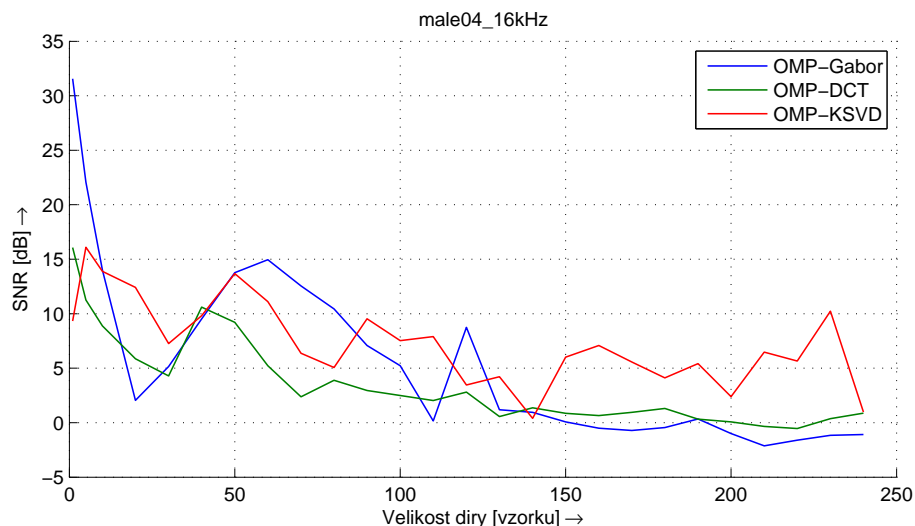
Na obrázku 7.1 je průběh pro ženský hlas. Nejlepší rekonstrukce lze v tomto případě dosáhnout s Gaborovým slovníkem. Neočekávané výsledky vrací K-SVD slovník. Pro většinu velikostí úseků je aproximace K-SVD dokonce horší než DCT. To je vzhledem k očekávání lepšího výsledku od komplexnějšího přizpůsobeného slovníku překvapivé.



Obr. 7.1: Ženský hlas

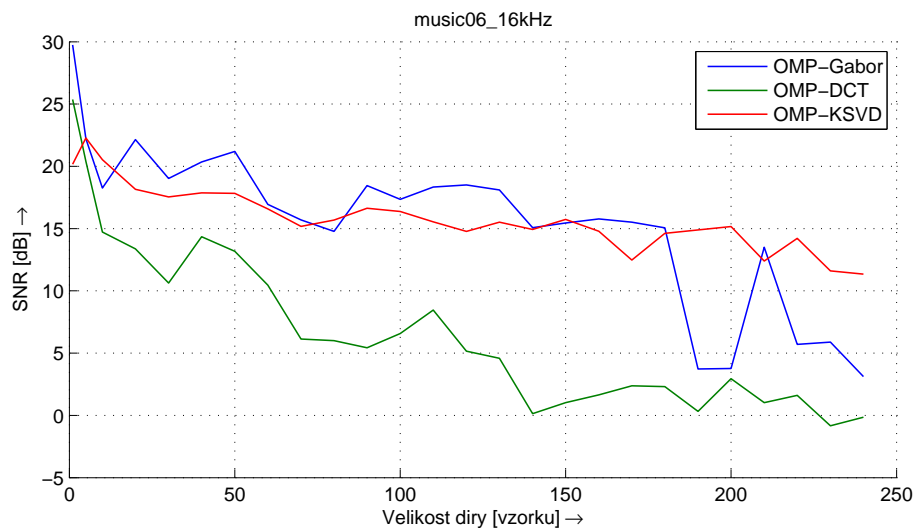
V dalším obrázku 7.2 je srovnání pro mužský hlas. Jak je možné vidět, od děr s velikostí 130 vzorků je K-SVD o pár decibelů lepší než oba statické slovníky. Zároveň lze na těchto místech pozorovat drobný náskok DCT. Dá se očekávat, že se s větším počtem informací ve slovníku (frekvence, fáze) mělo dosahovat lepších aproximací signálu. Podle získaných dat to je ve většině případech pravda, ovšem na některých místech vykazuje DCT lepší odstup než Gaborův slovník.

Jako zástupci hudebních signálů jsou vyhodnoceny `music06_16kHz.wav`, následně `music07_16kHz.wav` a nakonec `music11_16kHz.wav`. Společně pro hudební



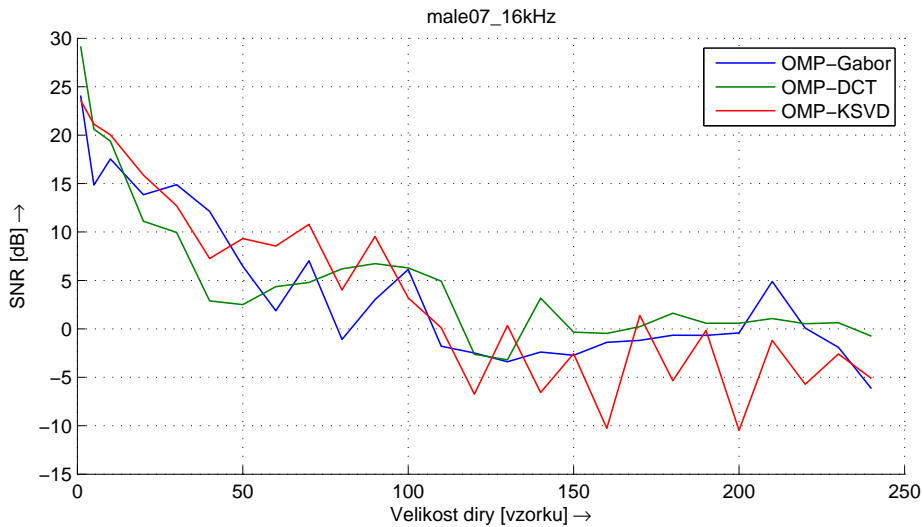
Obr. 7.2: Mužský hlas

signály se chybějící úseky generují od pozice 33 000. vzorku. Signál `music06_16kHz` představuje pořad spíše ženský hlas, nyní ale zpěv. Charakteristiku kvality rekonstrukce je možné vidět na obrázku 7.3. Zde je K-SVD podobně účinný jako Gaborův a zároveň lze pozorovat jeho větší stálost. Nedochozí k prudkým výkyvům. Slovník DCT vychází tentokrát nejhůře.



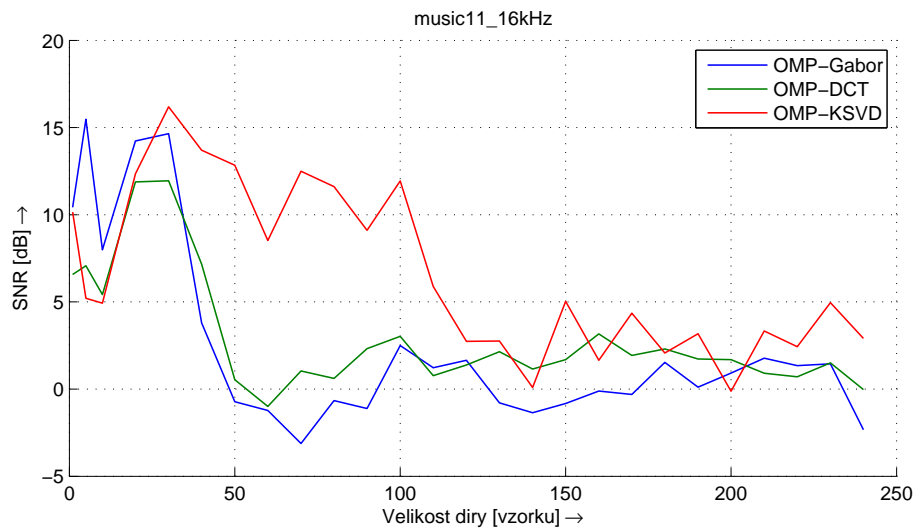
Obr. 7.3: Ženský zpěv

U dalšího souboru `music07_16kHz` se záznamem hry na bicí a charakteristikou na obrázku 7.4 je možné pozorovat velmi vyrovnané výsledky. Slovník DCT na hodně místech opět předčí Gaborův.



Obr. 7.4: Hra na bicí

Nakonec je na obrázku 7.5 zobrazeno srovnání ze souboru `music11_16kHz`. Signál představuje hru na kytaru. V první části (30. až 130. vzorek) vychází výrazně lépe K-SVD. Opět jsou velmi srovnatelné oba statické slovníky. Je dobré se tedy zamyslet, zda v aplikacích, kde bude požadováno především rychlé zpracování nevyužít radši DCT.



Obr. 7.5: Hra na kytaru

8 ZÁVĚR

V bakalářské práci bylo úkolem seznámení s technikami k optimalizaci slovníku pro rekonstrukci řídkého signálu a praktické srovnání jednoho optimalizovaného se zástupci nejznámějších statických slovníků. Ze širokého spektra používaných algoritmů k aproximaci byl vybrán a popsán OMP (Orthogonal Matching Pursuit). V kombinaci s OMP bylo dále voleno mezi statickými slovníky DCT a Gaborovým. Ve třetí kapitole byly tyto slovníky definovány.

Mezi různými metodami pro optimalizaci slovníků byly vybrány a popsány ty nejznámější případně svou funkcí nejzajímavější. Z nich byl vybrán k realizaci a porovnání právě algoritmus K-SVD. Čtvrtá kapitola proto detailně popisuje celý algoritmus od prvotního definování slovníkové matice přes hledání chybové matice až k jejímu rozkladu a následné aktualizaci slovníkových atomů.

Pátá kapitola pak popisuje implementaci v prostředí Matlab. Pro realizaci rekonstrukce je užitý Audio Inpainting Toolbox. K natrénování K-SVD slovníku je využíván K-SVD toolbox. Oba jako součást balíku SMALLbox. Tyto součásti byly upraveny a přetvořeny do podoby užitelné k reprodukci všech, v práci dosažených výsledků.

Dále byly řešeny optimální parametry k nastavení algoritmu pro učení K-SVD slovníku. Mezi tyto parametry patří posun segmentů, inicializační slovník, řídkost a počet iterací. Jednotlivá nastavení byla prakticky otestována a vybrána ta nejvhodnější. Nakonec bylo provedeno srovnání vybraných slovníků na různých zvukových signálech. Na určených místech byly do signálu generovány chybějící úseky představované nulovými vzorky a následně rekonstruovány za použití slovníku. Vyhodnocována byla kvalita rekonstrukce (představovaná odstupem původního a rekonstruovaného signálu) na různých velikostech chybějícího úseku.

Srovnání prokázalo ve většině případech lepší rekonstrukce za použití Gaborava oproti DCT slovníku. Ale lepších výsledků nebylo dosaženo vždy. U zpracování signálu s mužským hlasem a dvou hudebních signálů DCT navzdory své jednoduchosti Gaborův slovník předčil. Přizpůsobený slovník K-SVD vykazuje také poměrně překvapivé výsledky. Vzhledem k jeho komplexnosti byly očekávány výrazně lepší aproximace. Ve většině případech je K-SVD srovnatelný s Gaborovým. Na signálu se zvukem kytary vykazuje dobré výsledky pro menší chybějící úseky. Zvláštní je srovnání u ženského hlasu, kde je dokonce horší než DCT.

Nelze tedy jednoznačně říci, který slovník je nejlepší ale pokud je vyžadována rychlost vytvoření slovníkové matice, je vhodné použít některý ze statických. Při dostatku trénovacích dat a vysokém výpočetním výkonu je možné využít K-SVD algoritmus.

LITERATURA

- [1] ADLER, Amir, et al. *A constrained matching pursuit approach to audio declipping*. *ICASSP 2011* [online]. 2011, [cit. 2011-09-16]. Dostupné z: <http://hal.inria.fr/docs/00/55/70/21/PDF/2011-05-22_ICASSP_declipping.hal.pdf>.
- [2] ADLER, Amir, et al. *Audio inpainting* [online]. 2011, [cit. 2011-11-26]. Dostupné z: <<http://hal.inria.fr/docs/00/57/70/79/PDF/RR-7571.pdf/>>.
- [3] AHARON, Michal, Michael ELAD a Alfred BRUCKSTEIN. *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation* [online]. 2006, roč. 54, č. 11, s. 4311-4322, [cit. 2012-04-20]. DOI: 10.1109/TSP.2006.881199. Dostupné z: <http://www.cs.technion.ac.il/~elad/publications/journals/2004/32_KSVD_IEEE_TSP.pdf>.
- [4] BRUCKSTEIN, Alfred M., DONOHO David L., ELAD Michael. *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] ELAD, Michael. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer Science+Business Media, 2010, 376 s. ISBN 9781441970107.
- [6] GOLUB, Gene H. a Charles F. Van LOAN. *Matrix computations: Johns Hopkins series in the mathematical sciences*. 2. vyd. Michiganská univerzita: Johns Hopkins University Press, 1989, 642 s. ISBN 0801837723.
- [7] HRBÁČEK, Radek, et al. *Řídké reprezentace signálů: úvod do problematiky*. *Elektrorevue* [online]. 2011, [cit. 2011-10-11]. Dostupné z: <<http://elektrorevue.cz/cz/download/ridke-reprezentace-signalu-uvod-do-problematiky/>>.
- [8] JAN, Jíří. *Číslíková filtrace, analýza a restaurace signálů*. 2. rozšířené vydání. Brno: Nakladatelství VUTIUM, 2002. 427 s. ISBN 80-214-1558-4.
- [9] KOVÁR, Martin. *Maticový a tenzorový počet*. Skriptum k předmětu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008.

- [10] MAIRAL, Julien, et al. *Online Dictionary Learning for Sparse Coding*. Proceeding ICML '09, Proceedings of the 26th Annual International Conference on Machine Learning. [online]. 2009, [cit. 2012-05-12], ISBN 978-1-60558-516-1. DOI: 10.1145/1553374.1553463. Dostupné z: <<http://www.di.ens.fr/sierra/pdfs/icml09.pdf>>.
- [11] RAVELLI, Emmanuel, Gal RICHARD a Laurent DAUDET. *Union of MDCT Bases for Audio Coding*. *IEEE Transactions on Audio, Speech, and Language Processing* [online]. 2008, roč. 16(č. 8), 1361-1372 [cit. 2012-02-09]. DOI: 10.1109/TASL.2008.2004290. Dostupné z: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4648210>>.
- [12] SKRETTING, Karl a Kjersti ENGAN. *Recursive Least Squares Dictionary Learning Algorithm*. *Signal Processing, IEEE Transactions* [online]. 2010, roč. 58, č. 4, s. 2121-2130, [cit. 2012-05-12]. Dostupné z: <http://www.ux.uis.no/~karlsk/Skretting_RLS-DLA.pdf>.
- [13] SKRETTING, Karl a Kjersti ENGAN. *Image Compression Using Learned Dictionaries by RLS-DLA and Compared with K-SVD* *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* [online]. 2011, s. 1517-1520, [cit. 2012-05-12]. Dostupné z: <http://www.ux.uis.no/~karlsk/Skretting_icassp11.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AIT Audio Inpainting Toolbox

BP Basis Pursuit

DCT Diskrétní kosinová transformace – Discrete Cosine Transform

GLA Generalized Lloyd Algorithm

ILS-DLA Iterative Least Squares Dictionary Learning Algorithm

IRLS Iterative Reweighted Least Squares

LARS Least Angle Regression homothropy method

LS-OMP Least Squares Orthogonal Matching Pursuit

ODL Online Dictionary Learning

OMP Orthogonal Matching Pursuit

OLA Overlap-add

RMSE Odmocnina ze střední kvadratické chyby – Root-Mean-Square Error

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 Obsah elektronické přílohy

46

A OBSAH ELEKTRONICKÉ PŘÍLOHY

Na přiloženém CD je elektronická verze bakalářské práce ve formátu PDF. Dále se zde nachází upravený SMALLbox se součástmi nutnými k reprodukci dosažených výsledků. Adresářová struktura důležitých součástí přílohy je následující:

| | |
|------------------------------------|-----------------------------------|
| smallbox_1.9_mod/ | |
| /data/audio/wav/ | – použité zvukové soubory |
| /inpainting/source/ | – zdrojový adresář pro Inpainting |
| /toolboxes/AudioInpaintingToolbox/ | – funkce z AIT |
| /Experiments/InpaintingExperiment/ | – experiment ke srovnání slovníků |
| /Problems/ | – generátory chybějících vzorků |
| /Solvers/ | – metody k rekonstrukci |
| /Utils/ | – pomocné funkce |
| /dictionaries/ | – generátory statických slovníků |
| /evaluation/ | – funkce k výpočtu SNR |
| /toolboxes/CVX/ | – balík funkcí potřebný pro K-SVD |
| /toolboxes/KSVD/ksvdbox/ | – funkce k natrénování K-SVD |
| /toolboxes/KSVD/ompbox/ | – balík s OMP algoritmy |