Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava
Katedra Informatiky

Zmena identity rečníka

(diplomová práca)

Matúš Petruľák

Školiteľ: RNDr. Marek Nagy

Bratislava, 2010

Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Zmena identity rečníka

diplomová práca

Študijný program: Informatika
Odbor: 9.2.1 Informatika
Školiteľ: RNDr. Marek Nagy

Bratislava, 2010 Matúš Petruľák

Čestne prehlasujem, že som túto prácu vypracoval samostatne s použitím citovaných zdrojov.

.....

Poďakovanie

Chcem sa poďakovať môjmu školiteľovi Marekovi Nagyovi za jeho neoceniteľné rady a tiež mojej rodine za podporu.

Abstrakt

Autor:	Matúš Petruľák
Názov práce:	Zmena identity rečníka
Škola:	Univerzita Komenského v Bratislave
Fakulta:	Fakulta matematiky, fyziky a informatiky
Katerda:	Katedra informatiky
Školiteľ:	RNDr. Marek Nagy
Rok:	2010
Rozsah práce:	44 strán

Táto práca sa zaoberá zmenou hlasu. Zmena hlasu je proces, ktorý modifikuje reč povedanú zdrojovým rečníkom tak, aby znela ako reč povedaná cieľovým rečníkom. V práci je navrhnutý prístup k zmene hlasu, založený na unit selection. Od ostatných prístupov k zmene hlasu sa líši tým, že výsledný zvuk generuje z databázy, v ktorej sú uložené časti reči cieľového rečníka. Na vyhľadávanie v databáze zvukov sa používajú koeficienty MFCC, ktoré sú vypočítané zo zdrojovej reči. Práca následne porovnáva navrhovaný spôsob zmeny hlasu s existujúcim riešením založeným na pravdepodobnostnej lineárnej transformácii spektrálnej obálky. Výsledky ukazujú, že navrhovaný prístup dokáže zmeniť hlas zdrojového rečníka na hlas cieľového rečníka, pričom vo výslednom zvuku sa môžu prejaviť neželané javy.

Kľúčové slová: zmena hlasu, unit selection, identita, rečník

Abstract

Author:	Matúš Petruľák
Title:	Speaker's identity change
University:	Comenius University in Bratislava
Faculty:	Faculty of Mathematics, Physics and Informatics
Department:	Department of Computer Scoemce
Supervisor:	RNDr. Marek Nagy
Year:	2010
Number of pages:	44

This thesis deals with voice conversion. Voice conversion is a process which modifies the speech uttered by a source speaker so that it sounds like being uttered by a target speaker. The thesis proposes here a new approach to voice conversion based on unit selection. Unlike other methods, this approach generates the resulting sound from a database in which parts of the target speaker speech are stored. To search the database we use the source speaker MFCCs. Next, the thesis compares the proposed voice conversion method with an existing solution based on probabilistic spectral envelope linear transformation. The results show that the proposed approach can change the source speaker voice to target speaker voice, but the resulting sound may suffer from various unwanted artifacts.

Keywords: voice conversion, unit selection, identity, speaker

Obsah

Kapitola 1

Úvod

Tento dokument opisuje moje úsilie a poznatky, ktoré boli získané pri skúmaní predmetu diplomovej práce. Diplomová práca sa zaoberá *Zmenou identity rečníka*, alebo ináč povedané *Zmenou hlasu*, či *Konverziou reči*. V anglickej literatúre aj pod názvom *Voice conversion*, alebo *Voice morphing*.

V nasledujúcich odstavcoch tejto kapitoly sú uvedené ciele diplomovej práce a jej motivácia, ktorá viedla k vzniku tejto diplomovej práce.

1.1 Úvod do konverzie reči

Zmenou hlasu nazývame spôsob, pomocou ktorého modifikujeme charakteristiku reči zdrojového rečníka tak, aby mala charakteristiku nejakého zvoleného cieľového rečníka.

Pod pojmom charakteristika hlasu si môžeme predstaviť tri rozdielne veci. Po prvé je to farba hlasu, teda charakterizácia pomocou základného hlasivkového tónu, formatnov a iných frekvenčných charakteriskík hlasu [10]. Na druhej strane je prozódia reči. Čiže spôsob akým rečník rozpráva, ako narába s hlasom, ako mení intonáciu počas prejavu. [23]. Po tretie sú to výrazové prostriedky, ktoré rečník volí.

Zmenou hlasu môžeme dostať napríklad aj hlas, ktorý je od zdrojového viac odlišný (napríklad pri zmene pohlavia rečníka, alebo pri výraznej zmene veku rečníka - dieťa oproti dospelému človeku).

Ak chceme vykonať zmenu hlasu, potrebujeme vzorky prejavu reči zdrojového aj cieľového rečníka. Celkovú úspešnosť zmeny hlasu môžu ovplyvniť rôzne faktory,

ako napríklad kvalita záznamu, alebo či sú v prejavoch zastúpené rovnomerne všetky fonémy. Navyše pre niektoré techniky zmeny hlasu [41, 27] sú potrebné časovo zarovnané prejavy oboch rečníkov. Teda také prejavy, kde zvukový prejav jednej fonémy vieme identifikovať u oboch rečníkov. Takéto techniky môžeme nazvať aj *text-dependent*.

Vynára sa otázka, či je možné pomocou techník zmeny hlasu spraviť aj tzv. *cross-language* zmenu hlasu. Teda keď cieľový a zdrojový rečník nehovoria tým istým jazykom. V tomto prípade je omhono zložitejšie hladať časové zarovnania prejavov. Dôvodom je to, že niektoré fonémy sa v niektorých jazykoch vyskytujú a v iných nie. Navyše sa ukazuje, že prozódia reči je závislá na jazyku [13, 8, 6, 50].

Výsledky zmeny hlasu sa môžu hodnotiť objektívne, ale aj subjektívne, ako napríklad počúvaním. Počúvacie testy sa používajú na hodnotenie prirodzenosti prejavu a schopnosť rozoznať rečníka. Schopnosť rozoznať rečníka je definovaná ako schopnosť priradiť zmeneným hlasom k originálnemu hlasu, na ktorý bol zmenený hlas zmenený [27].

Základným procesom pri zmene hlasu je transformácia spektrálnej obálky zdrojového rečníka tak, aby sa zhodovala s obálkou cieľového rečníka. Boli navrhnuté rôzne systémy. Napríklad *codebook mapping* [3, 4], mapovanie formantov [21]. Tieto systémy avšak viedli k rôznym neprirodzeným artefaktom vo výslednom prejave [4]. Z toho dôvodu sú momentálne najpopulárnejšie metódy pozostávajúce z lineárnych transformácií, konkrétne prístup so spojitými pravdepodobnostnými transformáciami [41], ktorý poskytuje základ moderným systémom na konverziu reči. V tomto prístupe sa vstupné dáta klasifikujú pomocou *Gaussian mixture model* (GMM) a aplikuje sa na ne množina ováhovaných lineárnych transformácií. Tieto lineárne transformácie sa obykle určia z časovo zarovnaných dát pomocou metódy najmenšej chyby.

V tejto práci sa nebudeme zameriavať na *cross-language* zmenu hlasu. Zameriame sa na zmenu hlasu v slovenskom jazyku s dostupnými časovo zarovnanými dátami. Testovanie výsledkov uskutočním subjektívne a zároveň aj pomocou automatických testov [43].

1.2 Cieľ práce

Cieľom tejto diplomovej práce je preskúmať problematiku spojenú so zmenou hlasu. Navrhnúť vylepšenia existujúcich systémov, otestovať ich a dať prípadne

odpovede na otázky, ktoré sa vynoria počas skúmania problému.

Práca ďalej skúma použitie navrhovaného systému zmeny hlasu na slovenské vzorky hlasu. Väčšina doteraz navrhovaných systémov bola testovaná s anglickým, alebo nemeckým jazykom. Existujú síce publikácie týkajúce sa zmeny hlasu v českom prostredí, avšak môj systém pracuje odlišne od systémov navrhovaných v daných publikáciách [20].

V práci navrhujem vlastný spôsob zmeny hlasu. Môj spôsob je inšpirovaný kokatenačnou syntézou hlasu založenou na *Unit selection* [24]. Na základe zdrojového signálu predikujem priamo časti zvukového signálu cieľového rečníka, ktoré sa pospájajú, čim sa dosiahne zmena hlasu.

Môj spôsob porovnám s pravdepodobnostným prístupom zmeny spektrálnej obálky [41], ktorý uviedol Stylianou vylepšeným o residual prediction uvedenou Youngom [49].

V práci uvádzam aj prístupo VTLN [43], ktorý uviedol Suendermann, avšak neporovnávam ho s našim spôsobom.

Samozrejme prácu nie je možné uskutočniť bez istých teoretických znalostí. Preto sú v práci zahrnuté kapitoly 2 a 3, ktoré dávjú prehľad o použitých technikách.

1.3 Motivácia

Zaoberať sa problémom zmeny hlasu bez motivácie z reálneho sveta by nemalo zmysel. Avšak ukazuje sa, že techniky zmeny hlasu sú použiteľné vo viacerých oblastiach.

Veľký význam má zmena hlasu pri text-to-speech systémoch. Teda systémoch, ktoré používateľovi vedia prečítať nejaký text. V tomto prípade vývojárom stačí aby aplikácia vedela rozprávať jedným hlasom a pomocou techník zmeny hlasu dostaneme apolikáciu, ktorá rozpráva viacerými hlasmi.

Ďalej môžeme zmenu hlasu využiť v dabingu filomv a hudobnom priemysle. Diváci sa určite potešia ak sa im nejaký zahraničný herec prihovorí v ich rodnom jazyku.

1.4 Organizácia diplomovej práce

Diplomová práca je organizovaná nasledovne: Kapitola 2 zhŕňa základné teoretické poznatky a techniky, ktoré budeme využívať pri práci s rečou. V nasledujúcej kapitole 3 ukážeme možné prístupy ku zmene hlasu a aj náš navrhovaný spôsob. Ďalej nadviažeme kapitolou 4, v ktorej bude presne uvedený systém na konverziu reči ktorý budeme skúmať a návrh testov. Výsledky experimentov s naším systémom sú uvedené v kapitole 5 a prácu je zakončená záverom v kapitole 6.

Kapitola 2

Teoretické základy

Aby sme mohli dobre navrhnúť systém na zmenu hlasu je dôležité porozumieť niektoým základom. Táto kapitola poskutuje úvod do základných vlastností rečového signálu. Popisuje spôsob produkcie reči, jeho matematický model a uvedie vlastnosti rešového signálu, ktoré sú charakteristikou rečníka. V ďalšej časti kapitoly sú uvedené vybrané algoritmy, ktoré sa vyskytnú pri samotnej zmene hlasu. V kapitole sa vyskytnú nasledovné pojmy:

- Model vytvárania reči
- Lineárna prediktívna analýza
- PSOLA
- DTW
- MFCC

2.1 Model vytvárania reči

Uvedieme stručný popis toho, ako sa reč vytvára človek podľa Psutku [37]. Základným zdrojom energie pre reč poskytuje dychové ústrojenstvo. Vydychovaný prúd vzduchu je z plúc odvádzaný priedušnicou, potom prechádza hrtanom a nadhrtanovými dutinami, kde sa modifikuje, a ako rečový signál je sa vyráža cez pery do okolitého prostredia.

Zvuk vzniká v hrtane, konkrétne v hlasivkách. Hlasivky kmitania vytvárajú periodický prúd vzduchových pulzov vnímaných ako zvuk. Frekvenciu kmitania hlasiviek nazývame aj frekvenciou základného hlasivkového tónu.

Následne prechodom hlasivkového tónu nadhrtanovými dutinami dochádza vplyvom rezonancie k rozloženiu akustickej energie vo vznikajúcom rečovom signále. Frekvencie okolo ktorých sa akustická energia sústredí nazývame formantové frekvencie, resp. formanty. Označujeme ich číslami od formantu s najmenšou frekvenciou $F_1, F_2, \ldots$ V prípade potlačneia niektorých frekvenčných oblastí hovoríme o antiformantoch. Vzniká zvuk tónového charakteru, ktorý tvorí základ znelých častí reči, hlavne samohlások. Rôzne zvuky teda vznikajú tak, že pohyblivé artikulačné orgány menia tvar nadrhtanových dutín.

Artikulátory prispievajú aj ku tvorbe neznelých zvukov. Vytvárajú prekážky pre vydychovaný vzduch a vzniká šum rôzneho druhu. Šum je základom neznelej časti reči ako sú napr. spoluhlásky.

2.1.1 Matematický model vytvárania reči

Model zdroja a filtra

Pri počítačocom zpracovávaní reči býva obvykle užitočné istým spôsobom modelovať procesy produkcie a vnímania reči a následne reprezentovať pomocou parametrov zvoleného modelu. Uvážime isté rozumné predpoklady, môžeme použiť lineárny, časovo invariantný model produkcie reči, ktorý však bude mať platnosť len pre krátke časové okamihy. V skutočnosti pre mnohé zvuky reči je možné počítať s tým, že typ budenia a vlastnosti hlasového traktu ostávajú takmer konštanté v časovom úseku 10-30 ms. Vďaka tomu sa môže zodpovedajúci model produkcie reči slkadať z lineárneho modelu hlasového traktu s pomaly sa meniacimi parametrami, ktorý je budený vhodným budiacim signálom. Signál budenia pritom predstavuje buď periodický sled pulzov s periódou T_0 pre znelú reč, alebo šumový signál pre neznelú reč. Uvedené vzťahy sú podľa Psutku [37].

Model hlasiviek

Frekvencia vibrácie hlasiviek závisí od tlaku vzduchy vydychovaného z pľúc, a vlastností hlasiviek. Rýchlosť kmitania hlasiviek môžeme potom modelovať ako výstup nízkopásmového filtra druhého rádu, ktorého parametre sú samozrejme rozdielne pre každého rečníka. Funkciu modelu hlasiviek môžeme uvažovať v tvare

$$G(z) = \frac{1}{[1 - exp(-cT)z^{-1}]^2},$$

kde c je neznámy parameter filtra a T je perióda vzorkovania.

Model hlasového traktu

Model hlasového traktu by mal rešpektovať hlavne vlastnosti hrdelnej, ústnej a prípadne aj nosovej dutiny. Parametre modelu závisia na tvare týchto dutín a sú poväčšine rozdielne pre každého človeka aj pri vyslovovaní toho istého zvuku. Rovnako sa významne menia i u rovnakého rečníka pri vyslovovaní rôznych zvukov (napríklad poybom jazyka). Takýto model by mal pozostávať z kaskády malého počtu dvojpólových rezonátorov. Frekvencie rezonátorov by mali zodpovedať frekvenciám jednotlivých formantov. Funkciu modelu hlasového traktu potom udávame v tvare

$$V(z) = \frac{1}{\prod_{i=1}^{K}[1 - 2exp(-\alpha_i T)cos(\beta_i T)z^{-1} + exp(-2\alpha_i T)z^{-2}]},$$

kde K je počet formantov, ktoré model rešpektuje (pre analýzu hovorenej slovenčiny obvykle postačuje K = 3 až 4 formanty), α_i a β_i sú neznáme koeficienty modelu a T je perióda vzorkovania.

Model vyžarovania zvuku

Prenosovú funkciu modelu vyžarovania uvažujeme v tvare

$$L(z) = 1 - z^{-1}.$$

Budenie modelu rečovej produkcie

Vstupom modelu rečovej produkcie je pre znelé zvuky sled pulzov s periódou T_0 a pre neznelé zvuky náhodný šum s plochým spektrom. Funkcia budenia je pri perióde vzorkovania T nenulová len v tých okamihoch, ktoré zodpovedajú perióde základného hlasivkového tónu T_0. Ak položíme periódu vzorkovania $T = 1$ bude z-obraz budiacej funckie vyzerať ako

$$U(z) = G\sum_{n=0}^{\infty}(z^{-b})^n = \frac{G}{1 - z^{-b}},$$

kde G vyjadruje parameter miery vybudenia vzhľadom na ku štandardnej jednotkovej úrovni. Zdroj náhodného šumu neznelých zvukov môže byť realizovaný napríklad generátorom náhodných čísiel.

Celkový model produkcie reči

Pre určenie celkového modelu produkcie reči potrebujeme zjednotiť jednotlivé modelu hlasiviek, hlasového traktu a vyžarovania do modelu s funkciou $H(z)$.

$$H(z) = G(z)V(z)L(z)$$

$$H(z) = \frac{(1 - z^{-1})}{[1 - exp(-cT)z^{-1}]^2 \left\{ \prod_{i=1}^{K} [1 - 2exp(-\alpha_i T)cos(\beta_i T)z^{-1} + exp(-2\alpha_i T)z^{-2}] \right\}}$$

Filter s uvedenou štruktúrou sa nazýva celopólový filter a zapisuje sa tiež v tvare

$$H(z) = \frac{1}{1 + \sum_{i=1}^{Q} a_i z^{-i}},$$

kde $Q = 2K+1$, K je počet formantov ktoré chceme filtrom (modelom) obsiahnuť.

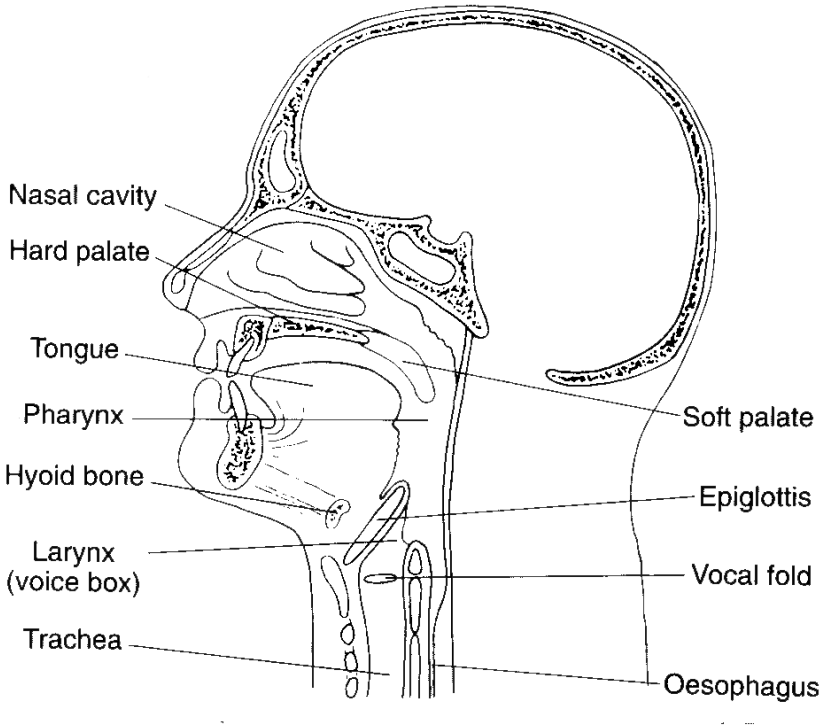
Na základe modelu budiaceho signálu (hlasiviek) a modelu rečovej produkcie vieme určiť výstup $S(z)$.

$$S(z) = H(z)U(z)$$

2.1.2 Reč ako postupnosť okienok

Ako bolo uviedené na predchádzajúcich riadkoch, na model produkcie reči sa môžeme pozerať ako na zdroj (hlasivky) a filter (ústna dutina, jazyk, pery, ...). Ďalej bolo uvedené, že budenie a aj vlastnosti hlasového traku sa nemenia príliš rýchlo a v časovom intervale 10-30ms ich vieme považovať za nemeniace sa. Čiže ak chceme zachytiť všetky dôležité informácie z rečového signálu je nevyhnutné rozdeliť si tento signál na malé časti, nazývané okienka. Avšak veľkosť okienka nemôžeme ľubovoľne zmenšovať. Základný hlasivkový tón u typyckého dospelého muža je v rozsahu 85 až 155Hz a u typickej dospelej ženy 165 až 255Hz [11]. Táto základná perodicita signálu nám určuje aké najmenšie okienko môžeme použiť. Použitie menšieho okienka by malo za následok že by sme zredukovali informácie v okienku tak, že potom by napríklad prestali fungovať metódy na základe lineárnej predikcie.

Pri rozpoznávaní a spracovávaní reči sa obyčajne používajú okienka, ktoré majú pevnú dĺžku. Okienka majú obyčajne dĺžku 10-30ms. Napríklad pri prenose hlasu mobilom (GSM) sa používa 20ms.



Obr. 2.1: Hlasový trakt [2]

Pri syntéze hlasu sa zvyknú používať okienka, ktoré majú svoju dĺžku závislú od veľkosti základného hlasivkového tónu. Táto vlastnosť umožňuje používanie štandarných techník na modifikáciu rýchlosti reči ako aj modifikáciu základného hlasivkového tónu. (napríklad PSOLA [9])

Vystáva otázka, ako rozdeliť signál na postupnosť okienok. Jeden spôsob je použiť laryngograf, ktorý nám presne zmeria hlasivkové pulzy. Ďalší spôsob je zrátať hlasivkové pulzy zo signálu. Ako najvhodnejší sa ukazuje byť algoritmus Praat [30, 7], ktorý navyše označí aj znelé a neznelé časti textu. Teda tie časti reči, ktoré vznikli periodickým budením hlasového traktu a tie ktoré nie.

2.2 Lineárna prediktívna analýza

Lineárne prediktívne (LP) kódovanie [32] je jednou z najefektívnejších metód analýzy akustického signálu. Je to metóda, ktorá sa snaží opäť na krátkodobom základe odhadnúť priamo z rečového signálu parametre modelu vytvárania reči, ktorého štruktúra bola ukázaná v časti 2.1. Metóda je atraktívna kvôli jej schopnosti zabezpečiť odhad uvedených parametrov pri relatívne prijatelnej výpočtovej záťaži.

Metóda je založená na modeli zdroja a filtra. Vstupom je postupnosť okienok. Pre každé okienko sa zrátjú LP koeficienty, ktoré čo najvernejšie predikujú daný signál. Rozdiel medzi skutočným signálom a predikovaným signálom nazývame *reziduá*. LP analýza nám teda zráta pre každé okienko filter a jeho budenie (reziduá). Keď reziduá prefiltrujeme získanými LP koeficientami, dostaneme pôvodný signál.

Navyše princíp LPC využíva predpoklad, že k-tú vzorku signálu ide popísať ako lieárnu kombináciu Q predchádzajúcich vzorkov a budenia $u(k)$, teda

$$s(k) = \sum_{i=1}^{Q} a_i s(k-i) + Gu(k),$$

kde G je koeficient zosilenia a Q je rád modelu. Funkciu $H(z)$ ide potom napísať v tvare

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{A(z)} = \frac{G}{1 + \sum_{i=1}^{Q} aiz^{-i}}.$$

Koeficienty a_i a G môžeme vyrátať napríklad pomocou Levinson-Durbinovej metódy [12]. Výpočet prebieha iterovane pre $i = 1, 2, \ldots Q$ ako je uvedený v algoritme 1.

Algorithm 1 Levinson-Durbin

$$\begin{aligned}
E_n^{(0)} &= R_n(0) \\
k_i &= -[R_n(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R_n(i-j)]/E_n^{(i-1)} \\
a_i^{(i)} &= k_i \\
a_j^{(i)} &= a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \qquad 1 \leq j \leq i-1 \\
E_n^{(i)} &= (1 - k_i^2) E_n^{(i-1)}
\end{aligned}$$

Hodnoty $E_n^{(i)}$ nazývame chybami predikcie. Navyše pre zosilenie G za predpokladu, že budiaca funkcia má tvar jednotkového impulzu alebo bieleho šumu platí

$$G^2 = R_n(0) + \sum_{i=1}^{Q} a_i R_n(i) = E_n.$$

Hodnota R_n je krátkodobá autokorelačná funkcia, pričom platí:

$$R_n(i) = \sum_{k=0}^{N-1-i} s_n(k) s_n(k+i)$$

Lineineárnu prediktívnu analýzu môžeme použiť na ďalšiu analýzu signálu. Pomocou LPC koeficientov môžeme určiť napríklad aj formantové frekvencie a základný hlasivkový tón.

Súvis LP koeficientov s zmenou hlasu

Ako uvidíme v ďalších kapitolách, pomocou zmeny LP koeficientov vieme dosiahnuť zmenu hlasu.

2.3 Ďalšie vlastnosti vlastnosti reči

V nasledovnej časti práce ukážeme ďalšie vlastnosti, ktoré môžeme z jednotlivých okienok vyextrahovať. Sú to:

- LSF
- MFCC

2.3.1 LSF

LP koeficienty môžeme alternatívne zapísať ako LSF (z anglického line spectral frequencies [25]. Koeficienty a_k z filtra $A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$ môžeme zapísať aj takto:

$$A(z) = \frac{1}{2}(P(z) + Q(z)),$$

kde

$$\begin{aligned} P(z) &= A(z) - z^{-(p+1)}A(z^{-1}), \\ Q(z) &= A(z) + z^{-(p+1)}A(z^{-1}). \end{aligned}$$

Z koreňov polynómu $P(z)$ a $Q(z)$, ktoré ležia na komplexnej jednotkovej kružnici dostaneme uhly $\omega_1, \omega_3, \ldots \omega_{P-1}$ pre $P(z)$ a $\omega_2, \omega_4, \ldots \omega_P$ pre $Q(z)$. Navyše tieto uhly môžeme ľahko meniť a neporušíme tým stabilitu filtra [40].

Ukázalo sa, že LSF majú vhodnejšie interpolačné vlastnosti ako iné reprezentície LP koeficientov [35].

2.3.2 MFCC

Ľudské ucho nevníma frekvencie lineárne. Melovské kepstrálne koeficienty [36] (MFCC z anglického Mel frequency cepstral coefficients) využívajú banku trojuholníkových pásmových filtrov s lineárnym rozložením frekvencií v tzv. melovskej frekvenčnej škále, ktorá je definovaná nasledovne:

$$f_m = 2595\, log_{10}(1 + \frac{1}{700}),$$

kde f [Hz] je frekvencia v lineárnej škále a f_m [mel] je tomu zodpovedajúca frekvencia v logaritmickej melovskej škále. Výpočet MFCC koeficientov prebieha pre každé okienko takto [48]:

1. Vykonáme Fourierovú transformáciu signálu.
2. Aplikujeme banku trojuholníkových filtrov (s lineárnym rozložením frekvencií v melovskej frekvenčnej škále).
3. Zlogaritmujeme výtsupy jednotlivých filtrov.
4. Vykonáme diskrétnu kosínusovú transformáciu (DCT z angl. discrete cosine transform) na výsledku z predchádzajúceho kroku.
5. MFC koeficienty sú amplitúdy výsledného spektra.

MFCC sa dajú použiť pri rozpoznávaní reči [44] a napríklad aj ako parametre signálu pri DTW. Algoritmus DTW uvádzame v sekcii 2.4.

2.4 Dynamic time warping

Niektoré algoritmy na zmenu hlasu požadujú, aby trénovacie dáta boli časovo zarovnané. Teda ak máme vektor príznakov, ktorý zodpovedá nejakému konkrétnemu hlasovému prejavu zdrojového rečníka, chceme identifikovať, ktorý vektor príznakov cieľového rečníka zodpovedá tomu istému hlasovému prejavu. Na časové zarovnanie prejavu dvoch rečníkov sa používa DTW (dynamic time warping).

Algoritmus DTW očakáva na vstupe dve postupnosti vektorov príznakov. Jednu pre zdrojového a jednu pre cieľového rečníka. Pomocou DTW vieme nájsť optimálne zarovnanie medzi dvoma zadanými postupnostami - algoritmus totiž zráta ich podobnosť. Algoritmus nájde riešenie pomocou metódy dynamického programovania. Výsledkom algoritmu je *best match* matica, ktorá pre každú podpostupnosť z prvej postupnosti hovorí, ktorá podpostupnosť z druhej postupnosti jej najlepšie zodpovedá. Na základe výstupu z DTW vieme určiť, ktoré časti postupností sa majú natiahnuť a ktoré naopak vymazať.

Algoritmus je uvedený ako Algoritmus 2 [1].

Algorithm 2 DTW

function DTWDistance(s[1..n], t[1..m])

for $i = 1$ to m **do**
  $DTW[0, i] \leftarrow infinity$
end for
for $i = 1$ to n **do**
  $DTW[i, 0] \leftarrow infinity$
end for
$DTW[0, 0] \leftarrow 0$
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $cost \leftarrow d(s[i], t[j])$
    $DTW[i, j] \leftarrow cost + min(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])$
  end for
end for
return DTW

2.5 PSOLA

Ako sme uviedli v časti 2.1.2 spracovávaný signál delíme na okienka. Pre každé okienko navyše vieme zrátať vektor príznakov (napr. MFCC, LPC). Teraz nastáva otázka, ako tieto okienka pospájať a dostať zvukový signál. Tu si môžeme pomôcť technikou zvanou PSOLA, ktorá je dobre známa pri syntéze reči.

PSOLA (z anglického Pitch-synchronous overlap and add) [9] je metóda, ktorá sa používa pri syntéze reči. Umožňuje jednoduchým spôsobom meniť výšku hlasivkového tónu F_0 a zároveň aj dĺžku prejavu. PSOLA má viac variant, z ktoých je najpopulárnejšia tzv. TD-PSOLA (time domain PSOLA) [19], najmä pre jej výpočtovú nenáročnosť.

Treba ešte poznamenať, že výšku hlasivkového tónu meníme len pre znelé časti reči.

Samotný algoritmus TD-PSOLA pozostáva z 3 krokov:

1. analýza
2. modifikácia
3. syntéza

Analýza

Počas analýzy sa signál rozdelí na krátke, prekrývajúce sa krátkodobé signály, ktoré majú dĺžku niekoľko (napr. 2) periód hlasivkového tónu. Majme hlasivkové pulzy (anlg. *pitch marks*)v časoch t_k. Rozložíme signál $s(n)$ na postupnosť krátkodobých signálov $s_k(n)$ tak, že

$$s(n) = \sum_{k=-\infty}^{\infty} s_k(n).$$

Pričom $s_k(n)$ získame z $s(n)$ takto:

$$s_k(n) = w_k(n - t_k)s(n),$$

kde $w_k(n)$ je obyčajne Hanningovo okienko dĺžky W_k definované ako:

$$w_k(n) = \begin{cases} 0.5[1 - cos(2\pi n/(W_k + 1))] & \text{pre } n = 0, 1, \ldots, W_k \\ 0 & \text{ináč} \end{cases}$$

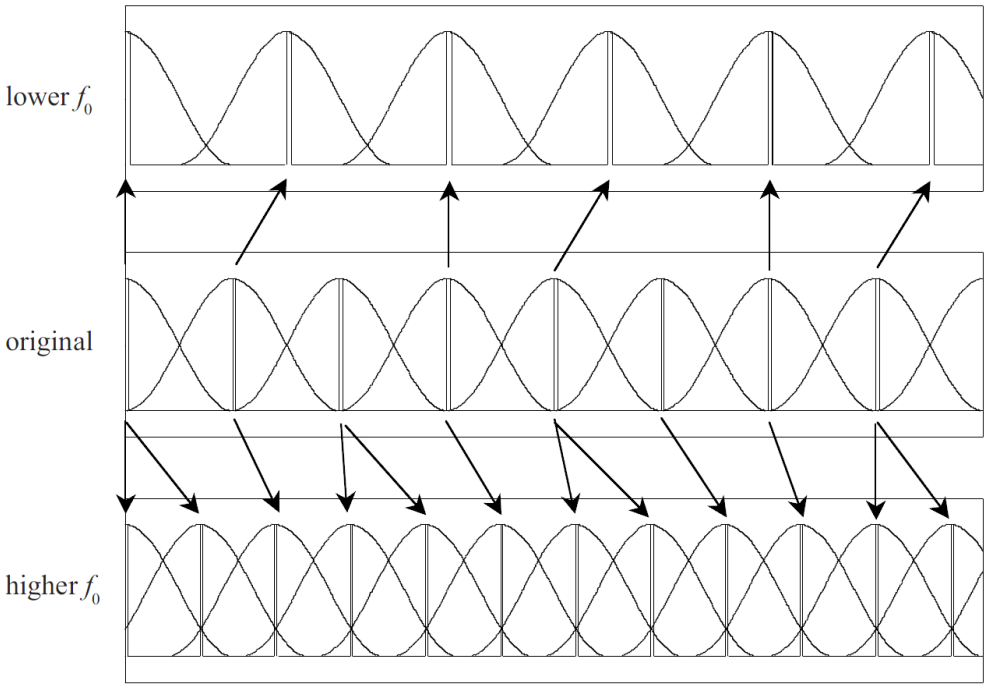
Hlasivkové pulzy vieme určit len pre znelé časti reči. Pre nezlené časti reči si vyrobíme pseudo hlasivkové pulzy, ktoré budú mať konštatnú dĺžku.

Modifikácia

Modifikácia predstavuje modifikovanie sledu hlasivkových pulzov. Dostaneme nové pulzy v čase t_l. K týmto pulzom sú následne priradené konkrétne krátkodobé signály. Zmena základného hlasivkového tónu F_0 sa dosiahne pomocou zmeny intervalov medzi hlasivkovými pulzami. Zmena tempa, alebo dĺžky prejavu sa dosiahne tak, že sa niektoré segmenty prejavy vynechajú, alebo sa znásobia. Príklad modifikácie je znázornený na obrázku 2.2.

Syntéza

Posledným krokom je syntéza reči. Počas tejto fázy sa vnikne rečový signál ako kombináciou jednotlivých krátkodových signálov, ktoré sú umiestnené na konkrétne pozície hlasivkových pulzov. Tieto pozície sa vyrátajú počas modifikačnej fázy.



Obr. 2.2: Príklad zmeny základného hlasivkového tónu f_0 pomocou PSOLA. Na obrázku je vidieť, ako sa jednotlivé okienka prelínajú a ako sa opakujú, resp. vymazávajú časti signálu. [43]

2.6 Zhrnutie

V tejto kapitole boli uvedené niektoré poznatky, ktoré budú užitočné pri samotnej zmene hlasu. Spomenuli sme, že reč pri spracovávaní delíme na okienka, ktoré sú synchronizované s hlasivkovými pulzami. Ukázali sme, že pre jednotlivé okienka si vieme zrátať vektory príznakov. Sem patria LPC, LSF a MFCC. Následne sme sa venovali algoritmu DTW, pomocou ktorého môžeme spraviť časové zarovnanie prejavov dvoch rečníkov. Algoritmus DTW na svoju prácu využíval vektory príznakov jednotlivých okienok. Ku koncu kapitoly sme uviedli algoritmus PSOLA, ktorý umožňuje jednoduchú modifikáciu základného hlasivkového tónu ako aj tempa, či dĺžky rečového prejavu. Algoritmus PSOLA navyše zachováva vysokú kvalitu a prirodzenosť hlasu.

Kapitola 3

Zmena identity reči

Táto kapitola popisuje základné prístupy používané na zmenu hlasu. V kapitole uvediem aj môj spôsob na zmenu hlasu založený na *unit selection* [24]. Zameriam sa na tieto prípstupy:

- Normalizácia dĺžky hlasového traktu
- Lineárne transformácie LPC
- Unit selection

Nasledovné riadky tejto kapitoly sa venujú detailnejšiemu popisu algoritmov.

3.1 Identita rečníka

Keď chceme uskutočniť zmenu identity rečníka pomocou zmeny hlasu, musíme si najprv rozobrať, ktorú vlastnosť hlasového prejavu chceme meniť. Hlasový prejav so sebou nesie viacero informácií, ktoré môžu byť charakteristické pre daného rečníka. Od niektorých základných vlastností, ako je výška hlasu (resp. výška základného hlasivkového tónu F_0), farba hlasu, ale aj to, ako rečník mení intonáciu a aké výrazové prostriedky volí.

Hlasový prejav ovsahuje tieto informácie [28, 43]:

- **Segmentové informácie** popisujú farbu hlasu, štruktúru formantov, F_0 a energiu akustického signálu. Tieto informácie sú zväčša dané stavbou hlasového traktu, ale môžu byť ovplyvnené napr. nádchou, alebo zmenou emočného rozpoloženia rečníka [29].

- **Supra-segmentové informácie** obsahujú informáciu o štýle rozprávania rečíka. Patra sem napríklad informácie o zmene intonácie (F_0), alebo o rýchlosti prejavu.

- **Linguistické informácie** zahrňujú voľbu výrazových prostriedkov, slov, prízvuk, nárečie, prípadne nejakú rečovú poruchy ako napríklad koktanie, zajakávanie.

V tejto práci sa zameriame hlavne na zmenu segmentových informácií. Tie sú totiž dané, ako sme uviedli, hlavne stavbou hlasového traktu. Človek dokáže ľahko ovplyvniť napríklad rýchlosť reči a začať hovoriť pomalšie či rýchlejšie, prípadne ovplyvniť intonáciu. Avšak väčšina ľudí, na rozdiel od imitátorov, nedokáže ovplyvniť stavbu svojho hlasového traktu, alebo formantovú štruktúru hlasu.

V práci nebudeme uvažovať supra-segmentové a linguistické informácie.

3.2 Algoritmy založené na modeli zdroja a filtra

Algoritmus zmeny identity reči je dosahuje zmenu identity rečníka modifikovaním parametrov akustickej reprezentácie rečového signálu. Obyčajne algoritmus zahŕňa dve časti, trénovaciu procedúru a následne samotné transformovanie reči transformačnou procedúrou.

Trénovacia procedúra pracuje s vzorkami reči zdrojového a cieľového rečníka. Zo vzoriek reči na vstupe sa získajú spektrálne parametre, ktoré reprezentujú identitu jednotlivých rečníkov. Tieto parametre obyčajne zachytávajú kátkodobé zvukové vlastnosti, ako tvar spektra a štruktúru formantov. Na základe týchto parametrov je natrénovaná transformačná fukcia, ktorá transformuje parametre zdrojového rečníka na zodpovedajúce spektrálne parametre cieľového rečníka.

Transformačná procedúra aplikuje natrénovanú transformačnú funkciu na spektrálne parametre zdrojového rečníka. Z takto získaných spektrálnych parametrov je nakoniec zosyntetizovaná reč.

To, aký algoritmus zvolíme na zmenu hlasu, určuje ktoré spektrálne parametre chceme zo vzoriek zvuku extrahovať, prípadne modifikovať.

Pri návrhu algoritmu treba zvážiť nasledovné oblasti:

3.2.1 Matematický model reprezentácie reči

Na to, aby sme mohli manipulovať s rečou v počítači, musí byť rečový signál reprezentovaný parametrami rečového modelu. Ukazuje sa, že na základe spektrálnej obálky je možné identifikovať rečníka počítačom [16, 17]. Aj z tohto dôvodu sa veľa systémov na zmenu hlasu zameriava na transformácie spektrálnej obálky a základného hlasivkového tónu. Jednou z možných reprezentácií je reprezentovať hlas pomocou modelu zdroja a filtra opísaných v sekcii 2.1.1. Parametre modelu sa zvyčajne vyrátajú pomocou lineárnej prediktívnej analýzy a dostaneme LP koeficienty (viď. sekcia 2.2). Tie je následne možné konvertovať na rôzne iné reprezentácie, napr. LSF (viď. sekcia 2.3.1). LPC reziduá dostaneme inverzným filtrovaním zdrojového signálu cez zodpovedajúci LPC filter. Keďže filter predstavuje hlasový trakt, inverzná filtrácia odstráni vplyv hlasového traktu zo signálu. Teda reziduá predstavujú budenie filtra, v našom prípade hlasivkové budenie.

3.2.2 Transformačná funkcia LPC

Význam tranformačnej funkcie je zachytiť vzťah medzi parametrami reči zdrojového rečníka a parametrami reči cieľového rečníka. Ak chceme správne natrénovať transformačnú funkciu, je nutné určiť, ktoré časti prejavu zdrojového a cieľového rečníka reprezentujú tie isté fonémy. Na dosiahnutie tohoto cieľa môžeme použiť algoritmus DTW (viď sekcia 2.4) [39]. Teraz, keď máme časovo zarovnané trénovacie dáta, budeme as venovať výbere transformačnej funkcie. Teda funkcie, ktorá vektor parametrov zdrojového rečníka prevedie na vektor parametrov cieľového rečníka.

Mapping codebooks a diskrétna transformácia

Medzi prvé metódy na transformáciu LPC patrila aj trasformácia pomocou kódovej knihy [3]. Pomocou vektorovej kvantizácie sa zostavila kódová kniha, ktorá priraďovala vektoru parametrov zdrojového rečníka iný vektor parametrov cieľového rečníka.

Ďalšia metóda rozdelila zdrojové vektory na niekoľko tried a pre každú triedu natrénovala transformačnú funkciu. Transformačná funkcia mohla byť napríklad vyrátaná pomocou lieárnej regresie [34].

Obe tieto metódy však vo vzniknutej reči vytvárali rôzne nespojitosti, ktoré vyplývali z diskrétnej povahy transformačných funkcií.

Spojité transformácie

V nasledujúcom texte uvediem príklad spojitej transformácie parametrov podľa Stylianou [41], založenej na pravpodobnostnom prístupe využívajúc *Gaussian mixture model* (GMM).

3.2.3 GMM

Využitím *Gaussian mixture model* (GMM) eliminujeme nevýhody diskrétnych transformácií. Jednotlivé vektory príznakov zdrojového rečníka totiž nezatrieďujeme do pevných tried, ale skonštruujeme GMM, ktorý modeluje rozdelenie zdrojových vektorov [41]. Tento prístup eliminuje tie nespojitosti, ktoré môžu nastať v predchádzajúcom modeli, keď vektor "preskočí" z jednej triedy do druhej. Navyše jednotlivé triedy zdrojových vektorov nie sú reprezentované len ich centroidmi, ale rovno celou pravdepodobnostnou distribúciou. Prístup založený na GMM sa ukázal byť aspoň tak efektívny ako predchádzajúce opísané prístupy [5].

Zmesový model (mixture model) nám umožňuje modelovať pravdepodobnostnú distribúci nejakého prvku x ako súčet, alebo zmes L komponentov (tried). V prípade GMM sú jednotlivé komponenty Gaussiány.

Predpokladajme teraz, že máme k dispozícii dve časovo zarovnané postupnosti vektorov príznakov (zvoľme si LSF reprezentáciu). Nech sú vektory n-rozmerné. (n zvolíme niekde okolo čísla 20). Zvoľme si označenie X pre zdrojové vektory a Y pre cieľové vektory. Najprv podľa [41] natrénujeme GMM na zdrojové vektory. GMM model bude mať L zložiek, pričom každá zložka bude reprezentovať jednu triedu foném. Na trénovanie použijeme *expectation-maximization* algoritmus [22]. Pre zmesový model GMM získame váhy jednotlivých komponentov α_i, stredné hodnoty μ_i a kovariančné matice Σ_i. Tieto parametre popisujú pravdepodobnostnú distribúciu

$$P(x) = \sum_{i=1}^{L} \alpha_i N(x|\mu_i, \Sigma_i), \quad \sum_{i=1}^{L} \alpha_i = 1,\ \alpha_i \geq 0,$$

kde $N(x|\mu, \Sigma)$ vyjadruje n-rozmerné normálne rozdelenie so strednou hodnotou μ a kovariančnou maticou Σ, pričom

$$N(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^L |\Sigma|}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right).$$

Podmienená pravdepodobnosť GMM triedy i za predpokladu i je podľa Bayesovho vzorca

$$P(i|x) = \frac{P(x,i)}{P(x)} = \frac{\alpha_i N(x|\mu_i, \Sigma_i)}{\sum_{j=1}^{L} \alpha_i N(x|\mu_i, \Sigma_i)},$$

kde $P(x,i)$ je pravdepodobnosť, že x je vygenerovaná i-tou triedou.

Transformačná funkcia

Transformačná funkcia vyzerá nasledovne: Funkcia dostane na vstupe vektor príznakov a jej výsutpom je transformovaný vektor príznakov. Funkcia je vyjadrená ako

$$\hat{y} = F(x) = \sum_{i=1}^{L} (W_i x + b_x) P(i|x),$$

kde W_i je transformačná matica a b_i je *bias* vektor triedy i.

Zostáva určiť transformačnú funkciu W_i a vektory b_i. Použijeme metódu najmenších štvorcov. Budeme hľadať také W_i a b_i, aby suma rozdielu štvorcov medzi $x \in X$ a $y \in Y$ bola čo najmenšia. Konkrétne minimalizujeme:

$$\sum_{i=1}^{|X|} |y_i - \hat{y}_i|^2$$

Z čoho dostávame:

$$\underset{W,b}{\operatorname{argmin}} \sum_{i=1}^{|X|} \left| y_i - \sum_{i=1}^{L} (W_i x + b_x) P(i|x) \right|^2.$$

Dá sa navyše ukázať [41], že W a b sa dajú vyjadriť ako lienárna kombinácia α, μ a Σ.

Reziduá

Algoritmus, ktorý je uvedený na predchádzajúcich riadkoch, uvažuje model zdroja a filtra. V našom prípade pomocou spojitej lineárnej transformácie využívajúcej GMM modifikujeme LSF, ktoré predstavujú parametre filtra, a dostaneme nové parametre filtra. S signálom zdroja, reziduami, sme nerobili nič.

Jedna možnosť ako získať výsledný zvuk je filtrovať pôvodné reziduá novým filtrom [26], čo však nevedie k dostatočnej zmene hlasu.

Ďalšou možnosťou je vyrábať reziduá ako jednotlivé hlasivkové pulzy, alebo biely šum (podľa toho, či a jedná o znelú, alebo neznelú časť reči) [47]. Tento prístup vedie k reči, ktorá znie umelo a neprirodzene.

Alternatívnou metódou je metóda *residual prediction* [49]. Základom tejto metódy je výber reziduí z prejavu cieľového rečníka.

Počas trénovacej fázy si zapamätáme všetky reziduá r_k a k nim zopovedajúce vektory príznakov y_k (LSF, alebo LPC). V transformašnej fázi k transformovanému vektoru $\hat{x}$ vyberieme taký vektor $\hat{r}$, ktorý spĺňa:

$$\hat{r} = r_{\hat{k}}, \text{ kde } \hat{k} = \underset{k=1,\ldots,K}{\operatorname{argmin}} |\hat{x} - y_k|$$

Táto metóda však môže vytvárať v reči rôzne neželané artefakty.

3.3 Normalizácia dĺžky hlasového traktu

Ako sme uviedli v predchádzajúcom texte, tvar hlasového traku významne ovplyvňuje identitu hlasu. Jedným z možných modifikácií hlasového traktu je, že zmeníme jeho dĺžku. Menenie dĺžky hlasového traku sa zvyčajne používa pri rozpoznávaní reči. V prípade rozpoznávania reči znormalizovanie dĺžky hlasového traktu vedie k lepšej kvalite rozpoznávania reči [38].

Normalizácia dĺžky hlasového traktu (ang. Voice tract length normalization - VTLN) sa zvyčajne vykonáva vo frekvenčnej oblasti. Na frekvenčnom spektre jednotlivých okienok sa vykoná deformačná (ang. *warping*) funkcia. Pri rozpoznávaní reči sa zvyčajne narába len s magnitúdou frekvenčného spektra. Fáza frekvenčného spektra pri ďalšom spracovávaní pri rozpoznávaní reči totiž nie je potrebná (napríklad pri výpočte MFCC v sekcii 2.3.2). Avšak pri zmene hlasu fáza spektra je potrebná na zachovanie prirodzenosti hlasu.

Deformačná funkcia

Uvažujme deformačnú funkciu h s parametrami $a_1, a_2, \ldots$ a nech $0 \leq \omega \leq \pi$ je spojitá normalizovaná frekvencia. Potom

$$\hat{\omega} = h_{a_1,a_2,\ldots}(\omega) \text{ kde } 0 \leq \hat{\omega} \leq \pi.$$

Uvažujme $X(\omega)$, funkciu, ktoré nám povie veľkosť magnitúdového spektra daného okienka v závislosti od zvolenej frekvencie ω. Túto funkciu môžeme deformovať a dostaneme $\hat{X}(\hat{\omega})$ s analogickou výpovednou hotnotou. Avšak magnitúda frekvencie $\hat{\omega}$ je rovnaká ako magnitúda ω, z čoho dostávame $\hat{X}(\hat{\omega}) = X(\omega)$.

Veľa známych funkcií z literatúry obsahuje len jeden parameter, zvyčajne označený α. Jednoduchá deformačná funkcia môže vyzerať napríklad takto: $g_\alpha(\omega) = \alpha\omega$.

Uvediem niektoré deformačné funkcie [43]:

- **piece-wise linear** $g_1(\omega, \alpha) = \begin{cases} \alpha\omega & \text{pre } \omega \leq \omega_0 \\ \alpha\omega_0 + \frac{\pi - \alpha\omega_0}{\pi - \omega_0}(\omega - \omega_0) & \text{pre } \omega > \omega_0 \end{cases}$

  - asymetric $\omega_0 = \frac{7}{8}\pi$

  - symetric $\omega_0 = \begin{cases} \frac{7}{8}\pi & \text{pre } \alpha \leq 1 \\ \frac{7}{8\pi} & \text{pre } \alpha > 1 \end{cases}$

- **power** $g_2(\omega, \alpha) = \pi \left(\frac{\omega}{\pi}\right)^\alpha$

- **quadratic** $g_3(\omega, \alpha) = \omega + \alpha \left(\frac{\omega}{\pi} - \left(\frac{\omega}{\pi}\right)^2\right)$

3.3.1 Zmena hlasu pomocu VTLN

V predchádzajúcom texte bolo uvedené, že použitie VTLN značne zlepší rozpoznávanie reči. Teda pomocou VTLN sa dá zmeniť hlas aby bol ľahšie rozpoznaný rozpoznávačom, ktorý vie dobre rozpoznávať nejaký konkrétny hlas. VTLN sa dá teda použiť aj na zmenu hlasu.

Transformácia

Predpokladajme, že používame deformačnú funkciu h s jedným parametrom α. Transformáciu vykonáme na jednotlivých okienkach signálu nasledovnými krokmi [43].

1. Pomocou diskrétnej Fourierovej transformácie [46] vyjadríme frekvenčné spektrum signálu x. $X = \mathcal{F}(x)$.

2. Použijeme deformačnú funkciu h s parametrom α a z X získame deformované frekvenčné spektrum $\hat{X}$, aby platilo $\hat{\omega} = h_\alpha(\omega)$ a zároveň $\hat{X}(\hat{\omega}) = X(\omega)$.

3. Inverznou Fourierovou transformáciou získame opäť časovú reprezentáciu deformovaného signálu. $\hat{x} = \mathcal{F}^{-1}(\hat{X})$.

Trénovanie

Predpokladajme, že máme transformačnú funkciu h a chceme natrénovať jej parameter α tak, aby sme následne mohli vykonávať vyššie uvedenú transformáciu. V prípade zmeny hlasu predpokladáme, že máme na vstupe časovo zarovnané dáta od zdrojového aj cieľového rečníka. Pre nejaký zvolený parameter α vieme zrátať akumulovanú vzdialenosť medzi jednotlivými prejavmi pomocou zrátavania vzdialeností medzi zmenenými okienkami zdrojového rečníka a príslušnými okienkami cieľového rečníka. Navyše je vhodné zarátať len tie okienka, ktoré obsahujú znelý zvuk [43].

Výpočet parametru α vyzerá nasledovne:

$$\alpha = \underset{\alpha'}{\operatorname{argmin}} \sum_{m=1}^{M} v_m \left| Y_m - \hat{X}_m(\alpha') \right|^2$$

Parameter v_m označuje, či dané okienko obsahuje znelý zvuk. $|X|$ je definovaná ako:

$$|X| = \sqrt{\frac{1}{\pi} \int_0^\pi |X(\omega)|^2 d\omega}$$

Parameter α' je vyberaný z konečnej množiny. Napríklad pre *piece-wise* lineárnu deformačnú funkciu

$$\alpha' \in \{0.76, 0.78, 0.80, \ldots, 1.24\}.$$

3.4 Unit selection

V tejto časi uvediem môj prístup k zmene hlasu. Môj prístup je inšpirovaný konkatenačnou syntézou hlasu [24] a v niečom je podobný prísutpu *residual prediction* [49], alebo prístupu, ktorý navrhol Suendermann na reziduá [43].

Myšlienka navrhovaného spôsobu zmeny hlasu spočíva v tom, že zvuk, reprezentujúci konvertovaný hlas dostaneme pospájaním vybraných častí zvuku cieľového

rečníka, ktoré sme videli pri trénovaní. Počas trénovania si teda vytvárame databázu cieľových zvukov z ktorých pri konverzii vyberáme. Tieto zvuky následne algoritmom PSOLA pospájame. Na vyberanie cieľových zvukov požijeme Viterbiho algoritmus [15].

Trénovanie

Predpokladajme, že máme časovo zarovnané dáta. Ak nemáme, použijeme algoritmus DTW (sekcia 2.4). Pre každé okienko zdrojového zvuku zrátame vektor príznakov x (napríklad MFCC) a uložíme si ho do databázy aj s časťou zvuku príslušného okienka cieľového rečníka. Položky v databáze zvykneme nazývať *units*.

Konverzia

Počas konverzie si vstupný zdrojový zvuk rozdelíme na okienka a pre každé zrátame vektor príznakov y. V databáze vyhľadáme pre každé okienko čo najpodobnejšiu vzorku (v zmysle vzdialenosti medzi x a y) a dané zvuky pomocou PSOLA pospájame a dostaneme výsledný zvuk.

Avšak tento prístup vedie k rôznym neželaným artefaktom vo zvuku [43]. Potlačenie týchto artefaktov môžeme spraviť dvoma spôsobmi. Môžeme skúsiť signál vyhladiť. Vyhladením však strácame prirodzenosť a identitu reči. Alebo môžeme skúsiť vybrať iné vzorky zvuku z databázy. Ak by sa nám podarilo vybrať lepšie vzorky z databázy, odstránili by sme niektoré artefakty v signále (napríklad vo výslednom zvuku sú počuť nespojitosti) a mohli by sme zvuk menej vyhladzovať. Použijeme prístup z konkatenačnej syntézy reči, nazývaný *unit selection* [24].

Pri použití unit selection musíme zadefinovať dve ceny:

- **Cieľová cena** (*target cost*) $t_{target}(x_k, y_k)$ udáva rozdiel medzi x_k v databáze a y_k na vstupe.
- **Cena spojenia** (*concatenation cost*, alebo *join cost*) $t_{join}(x_{k-1}, x_k)$ udáva cenu, za ktorú môžeme vybrať po sebe x_k a x_{k+1} z databázy.

Následne z databázy pre vstup y vyberieme takú postupnosť vektorov x, ktorá minimalizuje súčet cieľových cien a aj cien spojenia.

$$x = \underset{x'}{\operatorname{argmin}} \sum_{k=1}^{m} [w_t t_{target}(x'_k, y_k) + (1 - w_t) t_{join}(x'_{k-1}, x'_k)],$$

pričom za x' volíme postupne všetky rôzne výbery z databázy dĺžky vstupu k. Váhou w_t môžeme ovplyvniť, či algorimus viac bude prihliadať na cieľové ceny, alebo ceny spojenia.

V praxi použijeme na hľadanie postupnosti x dynamické programovanie, konkrétne Viterbiho algoritmus [15]. Tento algoritmus výrazne zrýchlime orezávaním tak, že si budeme pamätať len konštantné množstvo najlepších hypotéz. Toto zrýchlenie nemá na kvalitu výstupu významný vplyv [43].

3.4.1 Cena spojenia a cieľová cena

Aby metóda unit selection fungovala s dobrými výsledkami, musíme vhodne zvoliť cenu spojenia a cieľovú cenu. Cieľová cena bude vzdialenosť medzi vektormi príznakov x v databáze a y na vstupe. Vzdialenosť zvolíme euklidovskú. Ako vektory príznakov x zvolíme MFCC koeficienty rádu 20.

Cenu spojenia zvolíme podľa Suendermanna [43]. V literatúre sa dajú nájsť rôzne ceny spojenia, napríklad založené na iných vzdialenostiach ako je euklidovská a iných vektoroch príznakov (napríklad LPC, LSF). Vepa [45] ukázal, že nie je významný rozdiel medzi použitím LPC,LSF a MFCC ani medzi rôznymi typmi vzdialeností.

Suendermannovu cenu spojenia sme si vybrali preto, lebo ostatné ceny spojenia (napríklad založené na euklidovskej vzdialenosti medzi vektormi MFCC) viedli k počutelne výrazne horším výsledkom, ako napríklad vynechávanie spojok, alebo menenie samohlások na koncoch slov.

Cenu spojenia teda definujeme ako štvorec euklidovskej vzdialenosti medzi normalizovanou reprezentáciou prislúchajúcich zvukových vzoriek určených v databáze vektormi x_{k-1} a x_k nasledovne:

$$t_{join}(x_{k-1}, x_k) = |R(w_{x_{k-1}}) - R(w_{x_k})|^2,$$

kde $R(w)$ predstavuje časovú (*time domain*) reprezentáciu zvuku w zmenšenú o jeho priemer $\bar{w}$ a normalizovanú na konštantú dĺžku a jednotkovú energiu:

$$R(w) = \frac{\mathrm{abs}(w - \bar{w})}{|w - \bar{w}|}$$

Navyše upravíme cenu spojenia tak, aby uprednostňovala výber tých dát, ktoré sa v trénovacích dátach objavovali po sebe. Vyrobíme aj abstratný vektor x_0 a priradíme ceny nasledovne: $t_{join}(x_0, x_1) = 0$ a $t_{join}(x_{k-1}, x_k) = 0$ ak vzorky x_{k-1} a x_k sa nachádzajú v trénovacích dátach po sebe.

3.4.2 Modifikácie

V prípade môjho návrhu zmeny reči si do databázy ukladáme celé časi zvuku cieľového rečínka a k nim prislúchajúce vektory príznakov zdrojového rečníka. Algoritmus môžeme modifikovať a v databáze si budem ukladať namiesto vektorov príznakov zdrojového rečníka, vektory príznakov cieľového rečníka. To nám okrem iného umožní rozširovať databázu bez toho, aby sme mali časovo zarovnané trénovacie dáta. Zostáva vyriešiť, ako pri konverzii reči dostaneme z vektorov príznakov zdrojového rečníka vektory príznakov cieľového rečníka. Na to použijeme pravdepodobnostnú lineárnu transformáciu založenú na GMM opísanú v časti 3.2.3 tejto práce.

Táto modifikácia predstavuje prípadnú možnosť ďalšej práce a v tejto práci sa ňou nebudem zaoberať.

3.4.3 Vyhladzovanie

Po vybratí prvkov z databázy sa môže stať, že vo výslednom zvuku sa stále objavujú neželané artefakty, nespojitosti. Tieto nespojitosti môžu byť spôsobené nedostatočnou veľkosťou databázy, alebo nedokonalosťou funkcií cieľovej ceny a ceny spojenia. Aby sme potlačili nespojitosti vo výslednom signále, môžeme signál vyhladiť. V tejto práci použijem techniku lineárneho vyhladzovania [14]. Z signálu, ktorý chceme vyhladiť zrátame LSF a k nim príslušné reziduá. Tieto LSF vyhladíme, ako aj k nim príslušné reziduá. Následne pomocou RELP syntézy získame vyhladený signál.

Lineárne vyhladzovanie

Ako bolo vyššie spomenuté, LSF majú vhodné interpolačné vlastnosti. V práci implementujem lineárne vyhladzovanie LSF tak ako ho navrhol Dutoit [14]. Základnou ideou lienárneho vyhladzovania je distribuovať rozdiel medzi susednými LSF, ktorý vznikol spojením dvoch častí zvuku.

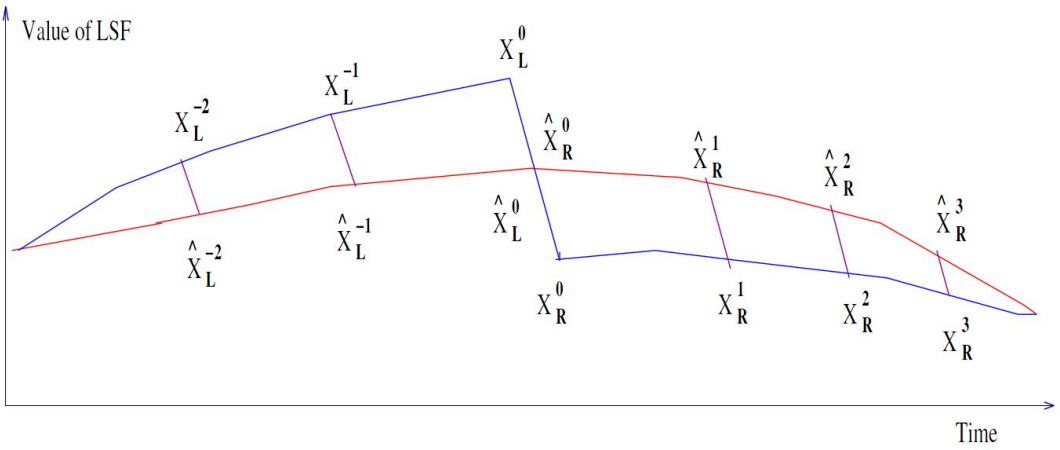
Nech X_L a X_R sú LSF pre okienka naľavo, resp napravo od miesta spojenia. M_L a M_R označujú do akej hĺbky chceme okienka naľavo, resp napravo ovplyvniť.

Nové LSF zrátame takto:

$$\hat{X}_L^{-i} = X_L^{-i} + (X_R^0 - X_L^0)\frac{M_L - i}{2M_L} \text{ pre } 0 \leq iM_L$$

$$\hat{X}_R^{-i} = X_R^{-i} + (X_L^0 - X_R^0)\frac{M_R - i}{2M_R} \text{ pre } 0 \leq iM_R$$

Lineárne vyhladzovanie ilustruje aj obrázok 3.1.



Obr. 3.1: Lineárne vyhladzovanie [45]

RELP

RELP (z angl. residual excited LPC) [31] je štandardná metóda na resyntézu zvuku. Zo zvuku sa získajú LPC a príslušné reziduá. Následne môžeme upraviť samostatne LPC alebo reziduá a následne späť zosyntetizovať zvuk.

3.5 Zhrnutie

V tejto kapitole boli ukázané tri prístupy k modifikáciu hlasu.

Prvý prístup bol založený na modeli zdroja a filtra. Modeloval filter pomocou LSF odvodených z LPC a následne tieto LSF transformoval využívajúc GMM. Tento prístup dokáže modifikovať spektrálnu obálku avšak nemení reziduá.

Druhý prístup sa opieral o metódu normalizácie dĺžky hlasového traktu. Pomocou priamej zmeny frekvenčného spektra dosahoval zmenu hlasu.

Tretí prístup bol založený na syntetizovaní hlasu z naučenej databázy. Bolo potrebné definovať ceny spojenia a cieľová cenu. Na výsledný výber prvkov z databázy bol použitý Viterbiho algoritmus. Získaný zvuk sa následne vhladil.

Kapitola 4

Metódy práce

V predchádzajúcich kapitolách boli uvedené prístupy, ktoré sa vyskytujú v súčasnej literatúre a prezentoval som môj spôsob. Cieľom tejto kapitoly je vysvetliť ako som pri práci postupoval, aké prostriedky som zvolil. Kapitola uvádza aj spôsoby vyhodnocovania a zberu dát, ktoré boli nevyhnutné k experimentom. Výsledky a ich interpretáciu prezentujeme až v kapitole 5.

4.1 Zmena hlasu

Popíšeme detailne dve metódy, ktoré budeme v práci používať na zmenu reči. Uvedieme v krokoch jednotlivé časti algoritmu, zvlášť pre trénovanie a zvlášť pre zmenu hlasu. Zvolíme tieto dva postupy:

(a) Transformácia spektrálnej obálky pomocou lineárnej transformácie LSF a zároveň využitie predikcie reziduí.

(b) Použiť navrhovaný prístup založený na algoritme unit selection.

Trénovanie aj transofrmácia má prístupy (a) aj (b) niektoré kroky spoločné a niektoré rozdielne. Rozdiely sú v jednotlivých bodoch naznačené ako alternatívy (a) a (b).

Trénovanie

Algoritmus na vstupe očakáva trénovacie dáta. Trénovacie dáta pozostávajú prejavu zdrojového a cieľového rečníka. Prejavy majú ovnakú obsahovú stránku (obaja

rečníci prečítajú tie isté vety). Viac o trénovacích dátach a ich získavaní budem ešte písať v časti 4.3.

Trénovanie vyzerá takto:

1. Pomocou algoritmu Praat určíme hlasovnové pulzy a označíme znelé a neznelé časti reči. Pre neznelé časti reči určíme pseudo-hlasivkové pulzy v konštatných rozostupoch.

2. Pracovať budeme s prekrývajúcimi sa okienkami. V strede okienka sa nachádza hlasivkový pulz a okienko je dlhé dve dĺžky hlasivkového tónu.

3. Vykonáme časové zarovnanie prejavov. Ako príznaky pre jednotlivé okienka použijeme MFCC koeficienty rádu 16.

4. (a) Zrátame LSF z LPC rádu 16 a natrénujeme lineárnu transformáciu LSF popísanú v sekcii 3.2.3. Počet GMM tried zvolíme 4. Toto číslo predstavuje počet umelých fonemických tried. Nemusí zopovedať reálnemu počtu foném v danom prejave.

   (b) Algoritmus nerobí nič.

5. (a) Uložíme si cieľové reziduá, ktoré sme videli počas trénovania a k nim príslúchajúce cieľové LSF.

   (b) Uložíme si cieľové časti zvuku, ktoré sme videli počas trénovania a k nim prislúchajúce zdrojové MFCC.

Transformácia

1. Pomocou algoritmu Praat určíme hlasovnové pulzy a označíme znelé a neznelé časti reči. Pre neznelé časti reči určíme pseudo-hlasivkové pulzy v konštatných rozostupoch.

2. Pracovať budeme s prekrývajúcimi sa okienkami, rovnako ako pri trénovaní.

3. (a) Zo signálu zrátame LSF. Urobíme transformáciu LSF. Dostaneme nové LSF, resp LPC.

   (b) Zo signálu zrátame MFCC. Predpočítame si matice cieľovej ceny a ceny spojenia pre Viterbiho algoritmus. Zvolíme váhu $w_t = 0,03$.

4. (a) Získame nové reziduá na základe predikcie reziduí podľa Ye a Young opísanej v sekcii 3.2.3.

   (b) Urobíme unit selection (viď. sekcia 3.4). Pomocou Viterbiho algoritmu vyberieme z databázy vzorky zvuku.

5. (a) Použijeme nové reziduá ako zdroj do filtra tvoreného novými LPC.
   (b) Rozdelíme signál na LSF a reziduá a obe lineárne vyhladíme. Následne spojíme cez RELP späť na signál.

6. Pospájame jednotlivé okienka pomocou PSOLA algoritmu. PSOLA algoritmus využijeme aj na dostiahnutie požadovanej výšky hlasivkového tónu.

Implementácia

Spomenutý algoritmus je implementovaný v prostredí MATLAB. Implementácia sa opeira Voice conversion toolbox for MATLAB [42], ktorý som rozšíril o náš spôsob zmeny hlasu. Niektoré časti algorimu som kvôli rýchlosti naprogramoval v jazyku C++. Alboritmus (a) som použil v nezmenej podobe v akej sa nachádzal vo Voice conversion toolbox for MATLAB. Algoritmus (b) som implementoval ja.

4.2 Metódy vyhodnocovania

Pri vyhodnocovaní systému na zmenu hlasu sa zamerim na tieto dve oblasti:

1. rozpoznateľnosť zmeneného hlasu, ako hlasu cieľového rečníka
2. kvalita zmeneného hlasu

Testy môžu byť buď subjektívne, alebo objektívne. Výhodou objektívnych testov je to, že sa dajú zvyčajne zopakovať a dostaneme rovnaký výsledok. Objektívne testy môžu byť vyhodnocované počítačom automaticky. Testy sa vyhodnocujú na základe parametrov zvuku, ktoré sa dajú z reči automaticky zrátať. Nevýhodou objektívnych testov je, že často nie sú zhodné s tým ako ľudia vnímajú zvuk. Navyše čo sa týka otázky kvality zvuku, neexistujú spoľahlivé objektívne metriky [43].

Subjektívne testy využívajú ľudí na to, aby ohodnotili výsledok. Nevýhodou subjektívnych testov je ich náročná opakovateľnosť a neobjektívnosť. Na druhej strane má význam robiť subjektívne testy, lebo práve ľudia budú tí, pre ktorých sa systémy na zmenu hlasu robia.

Uvediem testy niektré testy, ktoré sa dajú nájsť v literatúre, prípadne testy, ktoré budeme používať v tejto práci. Nasledovné testy sú určené na rozpoznateľnosť zmeneného hlasu.

4.2.1 Log-Spectral Distortion

Log-Spectral Distortion [18] (LSD) meria podobnosť dvoch paralelných prejavov. Na to, aby sa mohol pri vyhodnocovaní použiť tento test, musí byť prejav cieľového rečníka časovo zarovnaný s prejavom zdrojového rečníka, na ktorom bola vykonaná zmena hlasu. Na tento účel použijeme algoritmus DTW uvedený v sekcii 2.4.

LSD sa snaží reprezentovať podobnosť medzi prejavmi podobne, ako ju vníma človek. Je definovaná ako vzdialenosť medzi kepstrálnymi koeficientami zmeneného zdroja a cieľa.

$$D_{LSD} = \frac{10\sqrt{2}}{K\ln 10} \sum_{k=1}^{K} |\hat{x}_k - y_k|$$

Z definície D_{LSD} vyplýva, že čím sú si dva prejavy podobnejšie, tým je LSD menšia. Tento typ testu budem používať aj ja v tejto práci.

4.2.2 ABX

ABX [33] test je subjektívna metóda na porovnávanie dvoch audio záznamov. Testerovi sú poskytnuté 3 vzorky zvuku. Vzorka A, vzorka B a neznáma vzorka X. Je úlohou testera povedať, na ktorú vzorku sa vzorka X viac podobá. Vzorka A je zdrojový prejav, vzorka B je cieľový prejav a vzorka X je zmenený zdrojový prejav na cieľový. Vzorky A a B môžeme náhodne zamiešať, aby testeri nevedeli, ktorá vzorka je zdrojová a ktorá cieľová. Vyhodnocovanie testu je jednoduché. Ak väčšina testerov označila vzorku X za podobnú cieľovej považuje sa zmena hlasu za úspešnú. V opačnom prípade sa považuje zmena hlasu za neúspešnú.

Avšak autori niektrých článkov [26] poznamenali, že niektorí testeri nevedeli identifikovať hlas a mali dojem akoby na vzorke X bol hlas tretieho rečníka.

Jednou z variant testu ABX je tzv. rozšírený ABX [43]. V rozšírenom ABX teste má tester možnosť označiť, že vzorka X sa nepodobá ani na vzorku A ani na vzorku B. Tento typ testu budem používať aj ja v tejto práci.

4.2.3 AB

Ďalším typom subjektívneho testu je test AB. Testerovi sú poskytnuté 2 vzorky zvuku A a B. Úlohou testera je určiť, nakoľko sú su dané dve vzorky zvuku podobné. V tomto prípade môžeme napríklad z jednej inštancie ABX testu vytvoriť dve inštancie AB testu a to tak, že testerovi postupne dáme vzorky A, X a potom vzorky B, X. Test AB nebudem v tejto práci používať.

4.2.4 Kvalita reči

Test zameraný na kvalitu reči funguje nasledovne: Tester dostane jednu vzoku a má povedať aká je kvalita prejavu. Tester sa má zamerať či zvuk znie prirodzene, alebo nie. Nemodifikovaná reč by mala mať výbornú kvalitu. Následne sa vyhodnocuje parameter MOS (mean opinion score), ktorý predstavuje priemer jednotlivých odpovedí.

V tabuľke 4.1 sú naznačené možné odpovede testera pri teste kvality reči.

odpoveď	kvalita
5	excelentná
4	dobrá
3	priemerná
2	nízka
1	zlá

Tabuľka 4.1: Kvalita reči

4.3 Rečové dáta

Na to, aby mohol systém na zmenu hlasu fungovať, potrebuje ako vstup rečové dáta. Dieto dáta musia spĺňať nasledovné požiadavky:

1. Veľkosť by mala byť dostatočne veľká na to, aby dostatočne pokrývala priestor možných foném, ktoré môže algoritmus dostať počas konverzie na vstup.

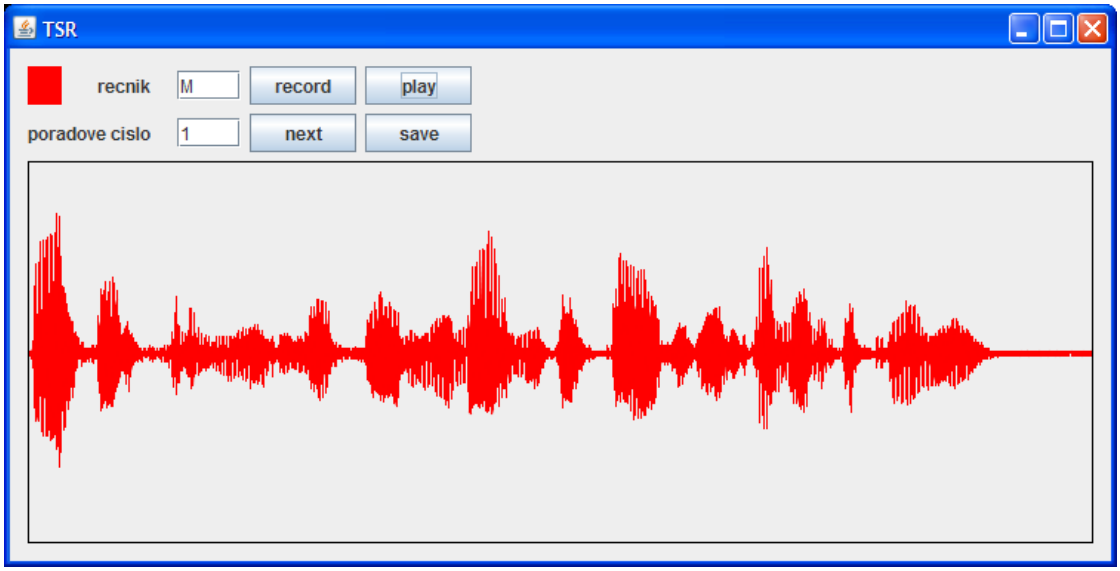
2. Dáta by mali pochádzať od viacerých mužských aj ženských rečníkov. Výsledky, ktoré získame na väčšej vzorke rečníkov dávajú lepší obraz o všeobecnej kvalite algoritmu.

3. Dáta musia byť časovo zarovnané, aby sme vedeli určiť prejavy rovnakých foném u zdrojového aj cieľového rečníka. Toto môžeme dostiahnuť napríklad tak, že rečníci povedia ten istý text.

Na základe požiadaviek bolo vybratých 19 viet. Všetky vety sa týkali jednej témy a jednotlivé fonémy sa v nich opakovali viac krát.

4.3.1 Nahrávanie

Vybral som 8 rečníkov vo veku od 19 do 25 rokov. Polovica rečníkov boli muži a polovica ženy. Vytvoril som aplikáciu (obrázok 4.1), ktorá uľahčila nahrávanie väčšieho množstva dát. Aplikácia automaticky odstránila ticho zo začiaku a konca nahrávky. Jednotlivé vety boli uložené do samostatných wav súborov. Použil som vzorkovaciu frekvenciu 16 kHz s 16 bitovým kódovaním. Pri nahrávaní som dával pozor, aby jednotlivé dáta neboli príliš prebudené alebo s príliš nízkym budením, čo by sa prejavilo v zhoršenej kvalite zvuku. Následne dáta boli normalizované po jednotlivých vetách programom Audacity. Použil som efekt *Normalize*, ktorý normalizoval maximálnu amplitúdu na -3 dB a signál vertikálne vycentroval.



Obr. 4.1: Aplikácia na nahrávanie zvuku

Kapitola 5

Experimenty a výsledky

V tejto kapitole uvádzam výsledky, ktoré boli namerané počas experimentov. Kapitola postupne uvádza výsledky pre subjektívne testy a potom pre objektívne.

5.1 Subjektívne testy

Z celkového počtu viet 19 pre každého z 8 rečníkov som náhodne vybral 5 viet do testovacej množiny a zvyšné som nechal v trénovacej množine. Vety z testovacej množiny neboli použité na trénovanie. Následne som náhodne zvolil 30 trojíc (zdrojový rečník A, cieľový rečník B, veta V z testovacej množiny). Pre každú túto trojicu som urobil 4 inštancie testov. Sú to:

1. Rozšírený test ABX. $A = V_A$, $B = V_B$, $X = V_{A'}$, kde $V_{A'}$ zmenená veta V_A pomocou algoritmu (a) na rečníka B.
2. Rozšírený test ABX. $A = V_A$, $B = V_B$, $X = V_{A''}$, kde $V_{A''}$ zmenená veta V_A pomocou algoritmu (b) na rečníka B.
3. Test na kvalitu zvuku V'_A.
4. Test na kvalitu zvuku V''_A.

Testy vyhodnocovalo 17 testerov cez webovú aplikáciu. Testeri sa dobre poznali so skupinou rečníkov. Dohromady testeri vykonali 1120 testov.

5.1.1 Výsledky

Podovnosť zvuku

Porovnával som algoritmus (a) s algoritmom (b). Porovnanie uvádzam sumárne, ale aj porovnávam samostatne konverziu z mužského rečníka na mužského, ženského na ženského a aj konverziu hlasu medzi pohlaviami.

Testeri označovali, či sa vzorka X podobá na viac na vzorku zdrojového rečníka A, alebo vzorku cieľového B, alebo ani na jedného z nich. Testeri nevedeli, akým algoritmom vznikla vzorka X, prípadne im boli rečníci A a B náhodne zamenení.

V tabuľke 5.1 sú zhrnuté výsledky pre algoritmus (a) a v tabuľke 5.2 sú zhrnuté výsledky pre algoritmus (b).

odpoveď	(a) spolu	(a) muž-muž	(a) žena-žena	(a) muž-žena	(a) žena-muž
zdroj A	35,00%	48,08%	45,61%	27,50%	27,47%
cieľ B	25,71%	13,46%	36,84%	32,50%	19,78%
žiaden	39,39%	38,46%	17,54%	40,00%	52,75%

Tabuľka 5.1: Výsledky testov ABX pre algoritmus (a)

odpoveď	(b) spolu	(b) muž-muž	(b) žena-žena	(b) muž-žena	(b) žena-muž
zdroj A	13,93%	7,69%	35,09%	11,25%	6,59%
cieľ B	58,21%	65,38%	38,60%	52,50%	71,43%
žiaden	27,86%	26,92%	26,32%	36,25%	21,98%

Tabuľka 5.2: Výsledky testov ABX pre algoritmus (b)

Ak by sme testovali ideálny algoritmus, tak by sa vzorka X takmer vo všetkých odpovediach testerov podobala na cieľ B. Z výsledkov vidno, že môj navrhovaný algoritmus (b) v testoch dopadol lepšie ako algoritmus (a). V 58% prípadov označili testeri vzorku vyrobenú mojím algoritmom za vzorku najviac podobnú rečníkovi B. V prípade algoritmu (a) to bolo len v 25% prípadov.

Kvalita zvuku

Porovnával som kvalitu zvuku algoritmu (a) a algoritmom (b). Z hodnotení testerov som urobil priemer a dostal som ukazovateľ priemerného skóre MOS. Výsledky sú nasledovných tabuľkách.

	spolu	muž-muž	žena-žena	muž-žena	žena-muž
MOS (a)	2,11	1,85	2,21	1,26	1,94
MOS (b)	1,08	0,95	1,48	0,69	1,24

Tabuľka 5.3: Výsledky testov kvality zvuku

Z výsledkov vidno, že kvalita reči vyprodukovaná mojím navrhovaným algoritmom testeri označili za výrazne horšiu. Náš algoritmus (b) celkovo dosiahol MOS 1,08 a algoritmus (a) 2,11.

Niektorí testeri tvrdili, že niektoré vzorky vyprodukované mojím algoritmom mali prijateľnú kvalitu a niektoré vzorky mali výrazne horšiu kvalitu. Tento fakt pripisujem tomu, že moja metóda na zmenu hlasu je veľmi citlivá na vhodné zvolenie parametru w_t. Teda parameter w_t nie je univerzálny pre všetkých rečníkov, ale treba ho určovať zvlášť. Zároveň som pri testovaní zistil, že parameter w_t má význam voliť len z obmedzeného rozsahu - ak ho totiž zvolíme mimo tohto rozsahu výsledok je skoro taký istý, ako by sme w_t zvolili 0, resp. 1.

5.2 Objektívne testy

Zvolil som objektívny test LSD (viď. sekcia 4.2.1). Pre každú možnú dvojicu zdrojový-cieľový rečník bola vykonaná zmenu hlasu na testovacej množine pomocou algoritmu (a) aj (b). Z zdrojovej reči X som získal zmenenú reč $\hat{X}_{(a)}$ a $\hat{X}_{(b)}$. Zmenenú reč som časovo zarovnal pomocou DTW s cieľovou rečou Y a zrátal som LSD. Výsledky ukazuje tabuľka 5.4.

	spolu	muž-muž	žena-žena	muž-žena	žena-muž
$D_{LSD}(\hat{X}_{(a)}, Y)$	5465	947	1594	1673	1250
$D_{LSD}(\hat{X}_{(b)}, Y)$	4970	821	1354	1597	1974
$D_{LSD}(X, Y)$	5295	876	1313	1946	1158

Tabuľka 5.4: Výsledky LSD

Môj algoritmus dopadol v LSD teste lepšie ako alboritmus (a). Pozoruhodné je, že po použití algoritmu (a) dostaneme zvuk, ktorý je zvuku Y menej podobný ako pôvodný vstup do algoritmu (a).

5.3 Zhrnutie

Na základe testov môžeme povedať, že môj systém na konverziu reči produkuje reč, ktorá obsahuje informáciu o zdrojovom rečníkovi, avšak kvalita zaostáva za modernými systémami na zmenu hlasu. Toto tvrdenie som podložil vykonanými testami.

V prípade algoritmu (a) sa mi nepodarilo priblížiť k výsledkom, ktoré dosiahol autor jeho implementácie. Ako možnú príčinu vidím to, že autor vyladil svoju implementíciu na anglický jazyk zvolením vhodných koeficientov a jeho implementácia mala problém so slovenčinou.

Testy navyše ukázali, že na testovacej množine môj algoritmus dosiahol omnoho lepšiu podobnosť ako algoritmus (a).

Kapitola 6

Záver

6.1 Zhrnutie

Táto práca sa zaoberala zmenou identity rečníka. V prvých troch kapitolách boli vysvetlené niektoré systémy na konverziu reči a navrhol som vlastný systém založený na *unit selection*. Môj systém som odskúšal a porovnal s iným systémom. Výsledky experimentov boli uvedené v kapitole 5.

6.2 Výsledky práce

V tejto práci sa mi podarilo dosiahnuť:

- Navrhlol som môj systém na zmenu hlasu založený na *unit seleciton*. Môj systém vyberá z databázy časti zvuku priamo na základe atribútov zdrojového signálu. V tomto prípade máme v databáze uložené časti hlasu cieľového rečníka. V tom je môj systém odlišný od systémov, ktoré v databáze majú uložené len reziduá (alebo niečo iné).
- Ukázal som, že môj systém je schopný generovať zvuk, ktorý poslucháči rozoznajú ako reč cieľového rečníka.
- Poukázal som na citlivosť parametra w_t pri mnou zvolených funkciách pre cieľovú cenu a cenu spojenia.
- Urobil som testy môjho algoritmu a iného algoritmu (a) popísaného v literatúre. Vo výsledoch som ukázal rozličné správanie algoritmov na zvolených

testoch. Môj algoritmus dosahoval výrazne lepšiu podobnosť s cieľovým rečníkom, na druhej strane algoritmus (a) dosahoval lepšiu kvalitu reči.

- Poukázal som na fakt, že jednotlivé algoritmy môžu podávať odlišné výsledky v závislosti od použitého jazyka.
- Implementoval som prostredie pre vhodné nahrávanie rečových prejavov, testovacie prostredie a ako aj samotný algoritmus na zmenu hlasu.

6.3 Možnosti ďalšej práce

Oblasti, v ktorých by sa navrhovaný algoritmus mohol zlepšiť sú kvalita reči a rozpoznateľnosť zvuku ako reči cieľového rečníka. Myslím si, že skúmaním funkcií cieľovej ceny a ceny spojenia by sa mohlo dôjsť k funkciám, ktoré by následne produkovali kvalitnejší a rozpoznateľnejší zvuk.

Keďže aj napriek lepším funkciám cieľovej ceny a ceny spojenia môže vo výslednom zvuku dochádzať k neželaným javom, navrhujem skúmať možnosti vyhladzovania signálu a odstraňovania neželaných artefaktov.

Ďalšiu z alternatív som navrhol v časti 3.4.2. Jedná sa o vyberanie častí zvuku z databázy na základe napríklad LSF cieľového rečníka. Pričom LSF cieľového rečníka môžeme získať pomocou lineárnej transformácie LSF zdrojového rečníka.

Literatúra

[1] Dynamic time warping - wikipedia. `http://en.wikipedia.org/wiki/Dynamic_time_warping`.

[2] Vocal tract. `http://www.ldc.upenn.edu/myl/lx1/sagittal1.gif`.

[3] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 655 –658 vol.1, apr 1988.

[4] Levent M. Arslan. Speaker transformation algorithm using segmental codebooks (stasc). *Speech Commun.*, 28(3):211–226, 1999.

[5] G. Bandoin and Y. Stylianou. On the transformation of the speech spectrum for voice conversion. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1405 –1408 vol.3, oct 1996.

[6] Alejandro Barbosa. A new mexican spanish voice for the festival text to speech system. Master's thesis, University of the Americas, Puebla, Mexiko, 1997.

[7] Paul Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345, 2001.

[8] S. Boudelaa and M. Meftah. Cross-language effects of lexical stress in word recognition: the case of arabic english bilinguals. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 121 –124 vol.1, oct 1996.

[9] F. Charpentier and M. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Acoustics, Speech, and Signal*

Processing, IEEE International Conference on ICASSP '86., volume 11, pages 2015 – 2018, apr 1986.

[10] D. Childers and C. Lee. Vocal quality factors: Analysis, synthesis, and perception. *Journal of the Acoustical Society of America*, 90(5), 1991.

[11] Mark Dolson. The pitch of speech as a function of linguistic community. *Music Perception*, 11(1), 1994.

[12] J. Durbin. The firring of time-series models. *Revue de l.Institut International de Statistique*, 28(3), 1960.

[13] Thierry Dutoit. *High Quality Text-To-Speech Synthesis of the French Language*. PhD thesis, Polytechnic Faculty of Mons, Mons, Belgium, 1993.

[14] Thierry Dutoit. *An introduction to text-to-speech synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[15] Jr. Forney, G.D. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268 – 278, march 1973.

[16] S. Furui. Research on individuality features in speech waves and automatic speaker recognition techniques. *Speech Communication*, 5:183–187, 1986.

[17] S. Fururi and M. Akagi. Perception of voice individuality and physical correlates. *Trans. Committee on Hearing Res., Acoust. Soc. Japan*, J66-A:311–318, 1985.

[18] R. Hagen. Spectral quantization of cepstral coefficients. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume i, pages I/509 –I/512 vol.1, apr 1994.

[19] C. Hamon, E. Mouline, and F. Charpentier. A diphone synthesis system based on time-domain prosodic modifications of speech. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 238 –241 vol.1, may 1989.

[20] Z. Hanzlíček and J. Matoušek. First steps towards new czech voice conversion system. In *System., 9th International Conference on Text, Speech and Dialogue, TSD 2006*, pages 383–390, 2006.

[21] C.-H. Ho, D. Rentzos, and S. Vaseghi. Formant model estimation and transformation for voice morphing. In *Proc. International Conference on Speech and Language Processing*, pages 2149–2152, 2002.

[22] R. Hogg, J. McKean, and A. Craig. *Introduction to Mathematical Statistics.* Prentice Hall, Englewood Cliffs, USA, 1995.

[23] Merle Horne. *Prosody: Theory and Experiment.* Kluwer Academic Publishers, 2000.

[24] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, pages 373–376, Washington, DC, USA, 1996. IEEE Computer Society.

[25] F. Itakura. Line spectrum representation of linear predictive coefficients. *Journal of the Accoustical Society of America*, 57(4):535–535, 1975.

[26] A. Kain and M.W. Macon. Spectral voice conversion for text-to-speech synthesis. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1, pages 285 –288 vol.1, may 1998.

[27] A. Kain and M.W. Macon. Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 2, pages 813 –816 vol.2, 2001.

[28] Alexander Kain. *High resolution voice transformation.* PhD thesis, Oregon Health and Science University, Portland, USA, 2001.

[29] D. H. Klatt and L. C. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *Journal of the Acoustical Society of America*, 87(2):820–857, 1990.

[30] B. Kotnik. First PMA/PDA evaluation campaign. In *Proc. of AST*, Maribor, Slovenia, 2006.

[31] D. T. Magill and C. K. Un. Speech residual encoding by adaptive delta modulation with hybrid companding. In *Proceedings of The National Electronics Conference*, pages 403–408, 1974.

[32] J. Markel and A. Gray. *Linear Prediction of Speech.* Springer, New York, USA, 1976.

[33] M. Meilgaard, G. Civille, and B. Carr. *Sensory Evaluation Techniques*. CRC Press, Boca Raton, USA, 1999.

[34] H. Mizuno and M. Abe. Voice conversion based on piecewise linear conversion rules of formant frequency and spectrum tilt. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume i, pages I/469 –I/472 vol.1, apr 1994.

[35] K. Paliwal. Interpolation properties of linear prediction parametric representations. In *Proc. of Eurospeech*, Madrid, Spain, 1995.

[36] J.W. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215 –1247, sep 1993.

[37] Josef Psutka, Luděk Muller, Jindřich Matoušek, and Vlastra Radová. *Mluvíme s počítačem česky*. Academia, 2006.

[38] D. Pye and P.C. Woodland. Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1047 –1050 vol.2, apr 1997.

[39] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[40] F. Soong and B. Juang. Line spectrum pair (lsp) and speech data compression. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, volume 9, pages 37 – 40, mar 1984.

[41] Y. Stylianou, O. Cappe, and E. Moulines. Continuous probabilistic transform for voice conversion. *Speech and Audio Processing, IEEE Transactions on*, 6(2):131 –142, mar 1998.

[42] David Suendermann. Voice conversion matlab toolbox. Technical report, Siemens Corporate Technology, Munich, Germany, 2007.

[43] David Suendermann. *Text-Independent Voice Conversion*. PhD thesis, Bundeswehr University Munich, Munich, Germany, 2008.

[44] V. Tyagi and C. Wellekens. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 1, pages 529 – 532, 2005.

[45] Jithendra Vepa. *Join Cost for Unit Selection Speech Synthesis*. PhD thesis, University of Edinburgh, 2004.

[46] J. Walker. *Fast Fourier Transforms*. CRC Press, Boca Raton, USA, 1996.

[47] C. Weinstein. A linear prediction vocoder with voice excitation. In *EASCON Proceedings*, 1975.

[48] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *PCM (3)*, pages 566–574, 2004.

[49] Steve Young and Hui Ye. High quality voice morphing. *Acoustics, Speech, and Signal Processing*, 1:9–12, 2004.

[50] Weibin Zhu, Wei Zhang, Qin Shi, Fangxin Chen, Haiping Li, Xijun Ma, and Liqin Shen. Corpus building for data-driven tts systems. In *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*, pages 199 – 202, sept. 2002.