Technická univerzita v Liberci

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B 2616 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídící systémy

Parametrizace vizuálního signálu řeči

Visual speech signal parameterization

BAKALÁŘSKÁ PRÁCE

Autor: **Vít Rychlovský**
Vedoucí bakalářské práce: Ing. Josef Chaloupka, Ph.D.
Konzultant Ing. Jindřich Žďánský, Ph.D.

V Liberci dne 12. 5. 2008

-------zde patří zadání bakalářské práce-----

Prohlášení

Byl jsem seznámen s tím, že se na mou bakalářskou práci plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce, či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum……

Podpis……

Abstrakt

Tato bakalářská práce se zabývá parametrizací vizuálního signálu řeči, konkrétně parametrizací izolovaných slov a jejich testováním. Práce je rozdělena na tři části. V první části jsou vysvětleny základní pojmy úlohy parametrizace a rozpoznávání. Dále jsou zde popsány DCT příznaky a geometrické příznaky získané z vizuálního signálu řeči a jednotlivé kroky parametrizace. V druhé části jsou uvedeny druhy metod rozpoznávání. Je zde vysvětlen pojem dynamické programování, neuronové sítě a popsána statická metoda skrytých Markovových modelů. Třetí část je zaměřena experimentálně. V této části je popsán postup trénování a rozpoznávání izolovaných slov za pomoci HTK Toolkitu, který je založen na metodě HMMs.

Klíčová slova: parametrizace, rozpoznávání, vizuální signál řeči, izolovaná slova, DCT příznaky, geometrické příznaky, HTK Toolkit, HMMs

Abstract

This bacalary work deals with visual speech signal parameterization, cocretely isolated words parameterization and their testing. The text is divited into three sections. In the first section, there are basic conceptions of parameterization and recognition problem explained. The DCT appearances and geometric appearances obtained from visual speech signal and individual steps parameterization are explained in this section too. In the second section there are recognition method kinds mentioned. The conceptions dynamic programming, neural networks and static method hidden Markov models are explainted in this section too. The last section is focused on some experiments. Traning and isolated words recognition by HTK Toolkit, which is based on HMMs, is describe in this section.

Keywords: parameterization, recognition, visual speech signal, isolated words, DCT appearances, geometric appearances, HTK Toolkit, HMMs

Obsah

Úvod

Už je to dlouho, kdy si lidé představovali budoucnost, kde počítače reagují na vyřčené povely a otázky. Tak daleko zatím nejsme, ale tato oblast výzkumu se velice rychle rozvíjí a to především díky rozvoji počítačové techniky. Systémy komunikující s člověkem mluveným slovem převážně rozpoznávají vyřčenou informaci pomocí akustického signálu. Mluvčí se ale může nacházet v prostředí zatíženém akustickým šumem. V tomto případě se využívá vizuální signál pro podporu akustického signálu. Celý proces je pak nazýván audiovizuálním rozpoznáváním. To by mohlo pomoci například lidem s postižením. Tyto systémy se už pomalu dostávají do praxe.

Cílem této bakalářské práce je provést vhodnou parametrizaci vizuálního signálu a následně otestovat parametrizované soubory v úloze rozpoznávání. Tato bakalářská práce tedy obsahuje postupy a metody parametrizace vizuálního signálu řeči. Dále obsahuje postupy trénování a rozpoznávání izolovaných slov. Ta jsou zpracovávána jednotlivými programy z programového souboru HTK Toolkit.

Tento projekt je možné použít při tvorbě většího systému využívajícího akustický i vizuální signál k rozpoznání mluvené informace.

1. Parametrizace vizuálních příznaků

Úloha rozpoznávání spočívá v zařazování objektů reálného světa do tříd. Nejprve je třeba určit, podle jakého hlediska budou dané objekty posuzovány. To znamená určit veličiny, které je charakterizují. Musí se určit přesnost a frekvence s jakou budou měřeny, čili definovat časoprostorovou rozlišovací úroveň. Naměřená data jsou vstupními údaji pro rozpoznávání. Tato data jsou uspořádána do vektoru, který se nazývá *obraz*. Prostor všech obrazů se nazývá *obrazový prostor*. Úloha rozpoznávání se dělí na dvě části:

a) zpracování dat
b) klasifikaci, čili zařazení do tříd

Naměřená data se zpracovávají tak, aby daná klasifikační metoda byla schopná s nimi správně pracovat. Metody se rozdělují do dvou tříd:

1. příznakové metody
2. strukturální metody

Při použití *příznakové metody* se kvalitativně oceňují podstatné vlastnosti objektů. Jak už je z názvu patrné, čísla, která tyto vlastnosti vyjadřují, nazýváme *příznaky*. Objekt je většinou popsán více příznaky, které tvoří *vektor příznaků*, prostor všech vektorů příznaků se nazývá *příznakový vektor*. Pokud je rozpoznávaným objektem křivka (velikost nějaké veličiny závislé na čase), je určení příznaků v celku jednoduché. Nejjednodušší vektor příznaků lze získat ze vzorků veličiny v čase. Většinou se však obraz nejprve zpracovává a pak se určuje vektor příznaků. K určení příznaků lze použít: časovou střední hodnotu průběhu, plochu pod kvádrem křivky apod. Další možností je za příznaky zvolit koeficienty Fourierova rozvoje. Při vytváření příznaků z vizuální informace se za příznaky volí koeficienty diskrétní kosinové transformace nebo geometrické příznaky, kterými většinou jsou vzdálenosti dvou bodů, plocha nějakého útvaru nebo míra zaoblení. Jelikož je příznakový prostor metrický, je možno v tomto prostoru vyjádřit podobnost.

Syntaktickými metodami převedeme nejprve průběh na vhodný strukturální popis. Strukturálním popisem míníme posloupnost charakteristických tvarů části křivky. Lze je popsat např.: náběh, pokles, zákmit. Tyto uvedené tvary tvoří abecedu použitou k popisu, jejich posloupnost je *slovem*. V úloze rozpoznávání se hledá, do jaké třídy testované slovo patří. V praxi jsou však data skoro vždy zatížena šumem. Proto je třeba

hledat nejlepší přibližnou shodu s některou ze tříd. Tyto metody převádějí úloho podobnosti na hledání nejmenší vzdálenosti.

Stroj, který provádí klasifikaci se nazývá *klasifikátor*. Klasifikaci lze chápat jako přiřazení symbolu označujícího třídu každému vektoru příznaků nebo slovu. Klasifikátor lze nastavit dvěma způsoby

1. Analýzou problému a definováním rozhodovacího pravidla před klasifikací
2. vytvořením rozhodovacího pravidla s použitím objektů, jejichž správná klasifikace je předem známa.

Hovoří se o *nastavování klasifikátoru učením*. Množinu, u které je klafikace známa, nazýváme *trénovací množinou*. Pro optimální nastavení klasifikátoru by byla potřeba nekonečně velká trénovací množina. Pomocí konečné trénovací množiny lze obecně zajistit správnost klasifikace pouze v konečném počtu případů. Nekonečně velké trénovací množiny lze zastoupit použitím pravděpodobnostního popisu obrazů a tříd.

V úloze rozpoznávání řeči počítačem se využívá jak akustická tak i vizuální složka. Výsledkem produkce řeči člověkem je akustický signál, který můžeme slyšet a pohyb řečového ústrojí, který můžeme vidět. Získání informace z vizuální složky je mnohem obtížnější než získat informaci ze složky akustické. Je to dáno tím, že vizuální složku lze získat jen z pohyblivých částí mluvícího člověka, což jsou: rty, jazyk, zuby a tváře. V praxi se nejčastěji pro rozpoznávání využívá rtů. Samotný pohyb a tvar rtů nám neposkytuje dostatečné množství informací o pronášeném slově (písmenu, větě). Vizuální složka se využívá k podpoře složky akustické. Takzvané odezírání ze rtů používají hlavně sluchově postižení, ale i zdraví lidé v hlučnějším prostředí. V tomto případě je akustická složka zatížena šumem, proto si pomáháme tím, že pozorujeme rty a snažíme se analyzovat, co dotyčný říká. Tak jako v jiných úlohách i při audiovizuálním rozpoznávání řeči počítačem se vychází ze zkušeností lidských expertů na odezírání. Sluchově postižení dosahují úspěšnosti odezírání maximálně 60-80% v závislosti na podmínkách odezírání. Takto vysoké úspěšnosti se zatím v počítačovém rozpoznávání řeči z vizuální složky nedaří dosáhnout. Lidé totiž nerozpoznávají strojově. Používají mozek, který jim dovoluje využívat více jak jen vizuální složku. Lidé dokáží doplnit chybějící slovo ve větě, neboť chápou význam slov (celé věty). Jako základní řečové jednotky se v úloze odezírání používají vizémy. Vizém je skupina fonémů, které mají podobný řečový obraz (vizuální složku řeči). Do jednoho vizému lze zahrnout fonémy p,b,m. Při mluvení dochází k tzv. ovlivňování jednotlivých vizémů

vyslovených v jenom slově (koartikulace). To je pro úlohu odezírání velký problém. Vizémy lze rozdělit na ovlivňované a ovlivňující. Mluvní obraz vizémů se může měnit podle toho, jak jednotlivé vizémi po sobě následují. Tento jev odezírání velmi ztěžuje. Analogií odezírání ze rtů je počítačové rozpoznávání na základě sledování oblasti rtů, což je jednou z kategorií vizuálních příznaků. Jednotlivé kategorie pro rozpoznávání řeči jsou:

- Tvarové vizuální příznaky
- Vizuální příznaky popisující informační obsah obrazu

Tyto jednotlivé příznaky se nejčastěji extrahují z obrazů, kde je mluvčí čelně natočen ke snímací kameře. Nověji se zkouší i extrakce vizuálních příznaků z aproximovaného 3D prostoru. Aproximovaný 3D prostor se vytváří buď ze snímků dvou kamer, které snímají mluvčího z dvou různých úhlů, nebo se použije jedna kamera a vhodně umístěné zrcadlo (popř. více zrcadel) a mluvčí je poté zaznamenán v jednom video snímku z dvou (i více) různých úhlů. Při pořizování obrazů je důležitá jejich kvalita (úhel pohledu, osvětlení). Nejvíce však záleží na řečníkovi. Ten by se měl snažit dobře artikulovat. Jak už bylo řečeno, prvními vizuálními příznaky jsou tvarové vizuální příznaky. Ty obvykle nazýváme geometrickými příznaky. Mezi nejpoužívanější geometrické příznaky patří: horizontální a vertikální rozšíření rtů, velikost oblasti rtů a zaokrouhlení rtů. Další tvarové příznaky lze vytvořit z analyzované hranice obrysu rtů. Buď se horní a spodní část aproximuje vhodnou křivkou nebo se sleduje směr hrany obrysů rtů. Podmínkou úspěšného a správného nalezení hranice rtů je to, aby rty měly odlišnou barvu než kůže, jazyk a zuby. Obraz také nesmí být rušen šumem a musí být bez poruch. Toto ale není vždy splněno, a proto se od této metody ustupuje.

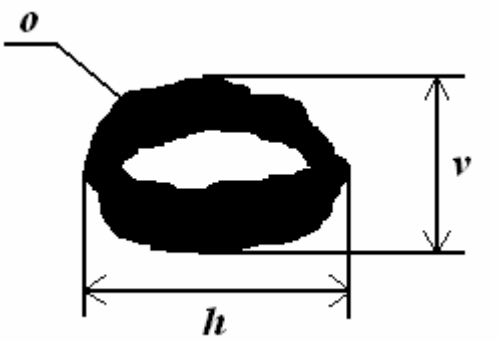
Dnes se nejvíce využívají vizuální příznaky, které popisují informační obsah obrazu. Počítačem zpracovávaná data jsou většinou digitalizované do čtvercového rastru. Obrazová funkce proto bývá reprezentována pravoúhlou celočíselnou maticí. Prvkem matice může být hodnota jasu nebo barevná hodnota obrazového bodu. Pokud by se měl vektor příznaků vytvářet například z obrazu o rozměrech 128 x 128, dostali bychom 16384 vizuálních příznaků pro jediný obraz (snímek). To je i pro výkonnou výpočetní techniku velké množství informace a vytvoření modelů například metodou HMM by bylo časově náročné. Je potřeba snížit objem dat popisujících obraz, ale bez

ztracení důležitých informací o obrazu. Proto se obraz transformuje vhodnou transformací a z tohoto transformovaného obrazu se vybírají složky (příznaky), které nejlépe popisují daný obraz. Běžně používanými transformacemi pro nalezení vizuálních příznaků jsou: diskrétní kosinová transformace DCT (the Discrete Cosine Transform, PCA (the Principal Component Analysis). Další transformací je diskrétní vlnková transformace DWT (the Discrete Wavelet Transform). Ta se ale téměř nepoužívá. V testech prováděných v této práci byly použity geometrické příznaky a DCT příznaky.

1.2 Geometrické příznaky

Pro získání geometrických příznaků je nutné mít k dispozici kvalitní snímky s dostatečně velkou oblastí zájmu. V této práci jsem použil audiovizuální databázi AVDBcz1. Snímky v této databázi tuto podmínku splňují.

Jako geometrické příznaky byly použity: horizontální *h* (1.1) a vertikální *v* (1.2) rozšíření rtů, dále oblast rtů *o* (1.3) a zaokrouhlení rtů *r* (1.4), které nabývá největších hodnot při vyslovování fonémů u, o, ř.



Obr. 1.1 Binární obraz oblasti rtů s vyznačenými geometrickými příznaky

$$h = \max_{y=0..M-1} \sum_{x=0}^{M-1} f(x,y), \tag{1.1}$$

$$v = \max_{x=0..M-1} \sum_{y=0}^{N-1} f(x,y), \tag{1.2}$$

$$o = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x,y), \tag{1.3}$$

$$r = v / h , \tag{1.4}$$

kde (x, y) jsou souřadnice obrazového bodu v binárním obraze nalezené oblasti rtů, $f(x, y)$ je obrazová funkce binárního obrazu nabývající hodnot 0, 1. M x N jsou rozměry binárního obrazu v obrazových bodech a max je funkce maxima. Jinou možností je použít příznaky získané z vnitřní oblasti rtů. V tomto případě je nutné, aby mluvčí dobře artikuloval. Dalším ztížením získání příznaků z vnitřní oblasti je i to, že jazyk a zuby mohou mít stejnou barvu, jako je barva kůže. Často se stává, že při segmentaci jsou body jazyku označovány za body rtů a dochází k získání nevhodných příznaků. Pro získávání všech příznaků ať už geometrických nebo příznaků z DCT transformace je nutné znát začátek a konec testovaného slova. Každé slovo z audiovizuální databáze je reprezentováno akustickým i vizuálním signálem. Průběh akustického signálu byl zobrazen na rovinu. Pak bylo možné určit, kde mluvčí začíná mluvit a kdy jeho projev skončil. Tím bylo zaručeno, že se příznaky získávají jen ze zkoumaného slova.

1.2.1 Normalizace geometrických příznaků

Velikost geometrických příznaků *h*, *v* a *o* je dána počtem obrazových bodů. Příznak r má relativní hodnotu. Příznaky *h*, *v* a *o* je nutné normalizovat, jelikož použití nenormalizovaných příznaků nevede k dobrým výsledkům. Je to způsobeno tím, že se rozměry oblasti rtů mohou měnit. Tyto příznaky byly normalizovány vydělením hodnoty příznaku jeho maximální hodnotou získanou z celého příznakového vektoru. Hodnoty příznaků *h*, *v* a *o* pak nabývaly hodnot 0 až 1. Hodnota příznaku *r* je již normována při výpočtu vzorcem (1.4).

1.3 DCT vizuální příznaky

Tyto příznaky jsou spolu s příznaky PCA dnes nejpoužívanější vizuální příznaky pro audiovizuální rozpoznávání řeči. Výhodou DCT příznaků oproti příznakům PCA je to, že se dají velice rychle vypočítat pomocí algoritmu rychlé kosinové transformace (FCT – Fast Cosine Transform). Velice důležité je při použití FCT, aby byl obraz nejlépe čtvercový a měl rozměr velikosti strany 2^n, kde *n* je celé kladné číslo. Pokud obraz tuto podmínku nesplňuje, musí se obraz na tuto velikost aproximovat. Ve většině případů se však obrazy vytvářejí už takové, že tuto podmínku splňují.

1.3.1 Vytvoření oblasti zájmu se rty pro FCT

Při analyzování mluvčích z videonahrávek databáze AVDBcz1, byla používána oblast zájmu o velikosti 128 x 128 obrazových bodů. Oblast zájmu je nutné pro využití DCT příznaků normovat. Objekt rtů by měl mít pro všechny mluvčí přibližně stejnou velikost a měl by se nacházet uprostřed oblasti obrazu zájmu. Nejdříve byl nalezen obličej, pak vybrána oblast zájmu se rty, ve které byla segmentací nalezena oblast rtů. Následně se z binárního obrazu zjistila šířka rtů h (1.1) a souřadnice těžiště x_t, y_t objektu rtů

$$x_t = \frac{m_{10}}{m_{00}}, \qquad y_t = \frac{m_{01}}{m_{00}}, \tag{1.5}$$

kde m_{pq} je obecný moment

$$m_{pq} = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} x^p y^q f(i, j), \tag{1.6}$$

kde (x, y) jsou souřadnice obrazového bodu v binárním obraze nalezené oblasti rtů, $f(x, y)$ je obrazová funkce binárního obrazu nabývající hodnot 0, 1. M x N jsou rozměry binárního obrazu v obrazových bodech. Dále je nutné obraz zmenšit nebo zvětšit. Pro zmenšení (zvětšení) obrazu s objektem rtů, bylo stanoveno, že výsledná šířka objektu rtů v obraze oblasti zájmu (128 x 128) je 100 obrazových bodů. Ze zjištěné šířky h objektu rtů byl poté určen koeficient zvětšení obrazu kz

$$kz = h/100 \tag{1.7}$$

Podle koeficientu zvětšení (zmenšení) byl původní obraz následně zvětšen (zmenšen). Dále bylo nutné použít geometrickou transformaci pro změnu měřítka. Se změnou měřítka byly zároveň přepočítány souřadnice těžiště

$$x_{tn} = x_t * kz, \qquad y_{tn} = y_t * kz, \tag{1.8}$$

kde x_{tn}, x_{tn} jsou nové souřadnice těžiště ve zvětšeném (zmenšeném) obraze a x_t, y_t jsou původní souřadnice těžiště. Podle nových souřadnic x_{tn}, x_{tn} byla vybrána ze zvětšeného (zmenšeného) obrazu oblast zájmu o velikosti 128 x 128 obrazových bodů.

1.3.2 2D Diskrétní kosinová transformace DCT

V této práci byl použit algoritmus 2D kosinové transformace FCT, vycházející z definice DCT-II (1.9). Tato transformace je pro zpracování obrazu využívána nejčastěji.

$$F(u,v) = \frac{2c(u)c(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos\left(\frac{2m+1}{2N} u\pi\right) \cos\left(\frac{2n+1}{2N} v\pi\right), \tag{1.9}$$

kde *f(m,n)* jsou hodnoty z původního obrazu o rozměrech *N* x *N* obrazových bodů, *F(u,v)* jsou koeficienty transformovaného obrazu, $0 \leq u, v \leq N-1$ a *c* jsou koeficienty (1.10).

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}} pro\ k = 0 \frac{1}{\sqrt{2}} \\ 1 pro\ k \geq 0\ \ 1 \end{cases} \tag{1.10}$$

1.3.3 Výběr a výpočet vizuálních příznaků z DCT

Použít celý transformovaný prostor DCT koeficientů jako vizuální příznaky je nepraktické, jelikož by jejich výpočet trval dlouho. K vlastnímu rozpoznávání slouží jen vybrané příznaky. Nejpoužívanější metody výběru příznaků jsou založeny na výpočtu energie E (1.11), rozptylu R (1.12) a normovaného rozptylu NR (1.13) z DCT koeficientů. Z těchto přepočtených koeficientů je poté vybíráno jako vizuální příznaky *P* koeficientů, které mají nejvyšší hodnotu.

$$E(u,v) = F(u,v)^2 \tag{1.11}$$

$$R(u,v) = \frac{(F(u,v) - \mu)^2}{N^2 - 1} \tag{1.12}$$

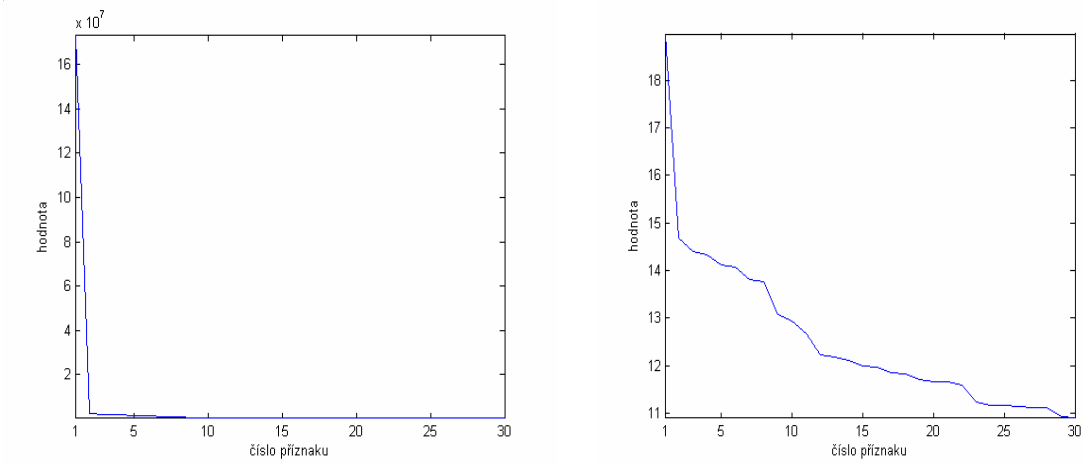
$$NR(u,v) = \frac{R(u,v)}{\mu^2}, \tag{1.13}$$

kde *F(u,v)* jsou koeficienty transformovaného obrazu, $0 \leq u, v \leq N-1$, *N* x *N* je rozměr obrazu a μ je střední hodnota vypočtená z koeficientů *F(u,v)* (1.14)

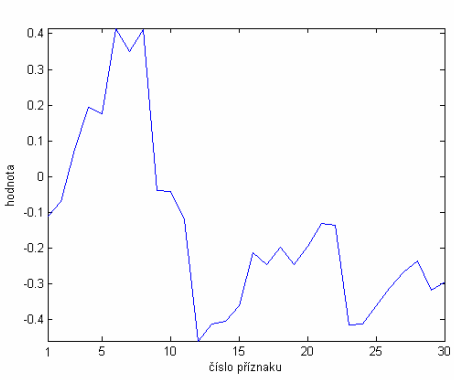
$$\mu = \frac{1}{N.N} \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} F(u,v) \tag{1.14}$$

1.3.4 Normalizace DCT příznakového vektoru

Z přepočtených koeficientů energie (6.11) je vybráno za vizuální příznaky *P* koeficientů, které mají nejvyšší hodnotu. Úrovně vypočtených hodnot DCT vizuálních příznaků se stoupajícím indexem razantně klesají, viz. obr 1.2. V této práci bylo použito pro potlačení tohoto jevu zlogaritmování všech hodnot z vizuálního příznakového vektoru. Další úpravou je odečtení střední hodnoty příznakového vektoru. Tím se eliminuje různá střední hodnota vizuálního příznakového vektoru z jednotlivých videonahrávek.



Obr. 6.5: Příznakový vektor tvořený třiceti nejvyššími hodnotami energie (vlevo) z DCT, zlogaritmovaný příznakový vektor (vpravo)



Obr. 6.6: Zlogaritmovaný příznakový vektor po odečtení střední hodnoty příznaků

1.4 Dynamické a akcelerační vizuální příznaky

Použití samostatných statických geometrických a vizuálních DCT příznaků, nevede k velkému rozpoznávacímu skóre. Z geometrických a vizuálních DCT příznaků

proto byly vypočítány příznaky dynamické a následně příznaky akcelerační. Tyto příznaky byly vypočítány pomocí jednoduché diference.

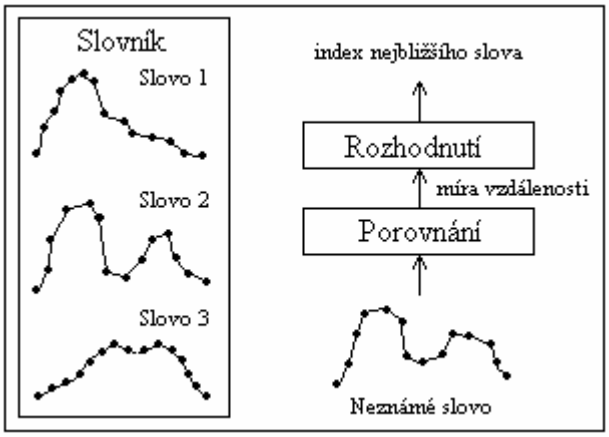
$$x'[n] = x[n] - x[n-1], \tag{1.15}$$

kde $x[n]$ jsou původní hodnoty vzorků, $x'[n]$ je vlastní diference.

2. Metody rozpoznávání

Počítačové rozpoznávání mluvené řeči je předmětem zájmu výzkumných laboratoří již téměř padesát let. Během této doby se vystřídalo několik metod pro rozpoznávání. Klasifikátory řeči lze podle použité metody rozpoznávání rozdělit na klasifikátory pracující na principu porovnávání se vzory a na klasifikátory pracující s využitím statických metod.

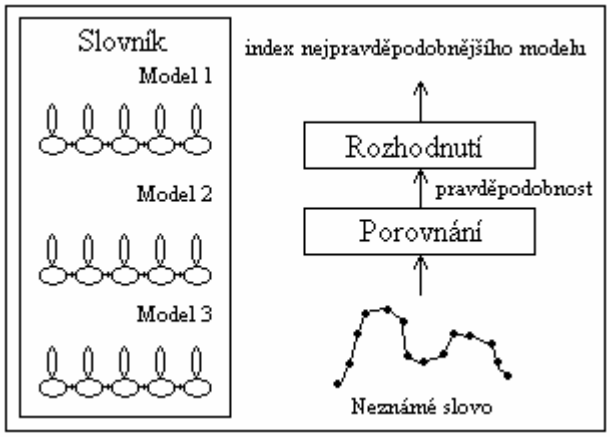
První skupina metod se převážně používala v sedmdesátých a osmdesátých letech. Byla aplikována zejména v klasifikátorech izolovaně vyslovených slov. Slovo je zpracováváno jako celek a je klasifikováno do té třídy, k jejímuž vzorovému obrazu má nejmenší vzdálenost. Hlavní úlohou je u těchto klasifikátorů určení vzdálenosti mezi dvěma obrazy slov. K určení této vzdálenosti se aplikuje metoda dynamického programování. U jednoho obrazu se nelineárně transformuje časová osa a oba obrazy se porovnávají, přičemž je snaha dosáhnout porovnání s co nejmenší výslednou vzdáleností. Algoritmus, který porovnává řečové obrazy na principu dynamického programování, pracuje s efektem nelineární časové normalizace. Kolísání v časové ose je modelováno časově nelineární funkcí (dynamic time warping – DTW).



Obr. 2.1 Klasifikátor pracující na principu porovnávání neznámého slova se vzory

V systémech druhé skupiny je přístup ke klasifikaci založen na statických metodách. Slova popřípadě celé promluvy jsou modelovány pomocí skrytých Markovových modelů (hidden Markov models – HMMs). Jednotlivá slova se buď modelují jako celek, anebo je model vytvořen zřetězením menších jednotek. Pro každou subslovní jednotku jsou stanoveny parametry Markovova modelu a neznámá promluva je rozpoznána na základě toho, jaká posloupnost slov, která je tvořená řetězcem modelů subslovních jednotek, generuje promluvu s největší pravděpodobností.

Mezi metody rozpoznávání patří také tzv. umělé neuronové sítě (artificial neural network – ANN). Tyto sítě jsou složeny z modelů neuronů a mají schopnost učit se. Celá síť se trénuje pomocí trénovaní množiny, přičemž se nastavují parametry jednotlivých neuronů tak, aby bylo dosaženo přesné klasifikace informací.



Obr. 2.2 Klasifikátor pracující na principu skrytých Markovových modelů

Technika rozpoznávání izolovaných slov založená na dynamickém programování se už nepoužívá. V současné době se nejvíce používají statické metody rozpoznávání, konkrétně metoda skrytých Markovových modelů. Tato metoda byla použita v této práci.

2.1 Dynamické programování a DTW

Při rozpoznávání mluvené řeči je správnost klasifikace negativně ovlivňována proměnlivostmi a nelineárním kolísáním signálu (akustického, vizuálního) v časové ose. Stejné slovo vyslovené týmž řečníkem nebude nikdy stejně dlouhé a ani jeho části nebudou stejně dlouhé. Záleží na fyzickém a psychickém stavu řečníka. Pokud pozorujeme slovo z akustického hlediska, délka jednotlivých částí, což jsou fonémy

nebo hlásky, se při každém vyslovení téhož slova mění. Při rozpoznávání řeči z vizuální složky se zaměřujeme na vizémy. Vizém je skupina fonémů, které mají podobný řečový obraz (vizuální složku řeči). Do jednoho vizému lze zahrnout například fonémy p,b,m. I vizémy nemají vždy stejnou délku, a proto lze použít metodu dynamického programování v obou případech. Nestejnou délku slov lze řešit lineární časovou normalizací. Ta však nedokáže postihnout časové kolísání uvnitř slova. S tímto problémem si dokáže poradit algoritmus, který je založen na dynamickém programování. Tento algoritmus pracuje s efektem nelineární časové normalizace. Kolísání v časové ose je modelováno časově nelineární funkcí (dynamické borcení času – dynamic time warping – DTW). Časové rozdíly, které vznikají mezi dvěma řečovými obrazy, jsou eliminovány nelineární transformací jedné z časových os. Je snaha dosáhnout maximální shody s druhým obrazem.

Při rozpoznávání pomocí DTW je jedno slovo rozpoznávané a druhé je použito jako referenční vzor. Tyto slova jsou vyjádřena svými obrazy *A* a *B*

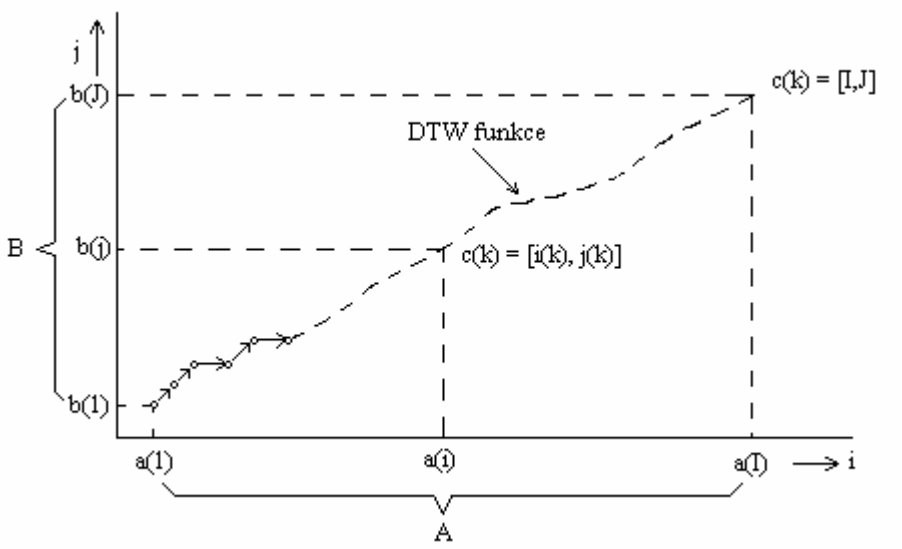
$$A = (\mathbf{a}(1),\mathbf{a}(2),\ldots,\mathbf{a}(i),\ldots,\mathbf{a}(I)),$$
$$B = (\mathbf{b}(1),\mathbf{b}(2),\ldots,\mathbf{b}(j),\ldots,\mathbf{b}(J)), \quad (2.1)$$

kde $\mathbf{a}(i)$ je *i*-tý vektor příznaků testovaného obrazu *A*, $\mathbf{b}(j)$ je *j*-tý vektor příznaků referenčního obrazu *B*. Algoritmem DTW se pak v rovině (*i*,*j*) hledá optimální cesta (obr. 3), tj. taková posloupnost bodů $\mathbf{c}(k)$ v této rovině

$$C = (\mathbf{c}(1),\mathbf{c}(2),\ldots,\mathbf{c}(k),\ldots,\mathbf{c}(K)), \quad (2.2)$$

kde $\mathbf{c}(k) = [i(k), j(k)]^T$, která minimalizuje funkci *D* celkové vzdálenosti mezi obrazy *A* a *B*

$$D(A,B) = \sum_{k-1}^{K} d\,[\mathbf{c}(k)]. \quad (2.3)$$



Obr. 2.3 Znázorně průběhu funkce DTW v rovině (*i*,*j*)

Přitom $d[\mathbf{c}(k)]$ je lokální vzdálenost mezi *i*-tým příznakem testovaného obrazu *A* a *j*-tým příznakem referenčního obrazu *B*. Při hledání optimální cesty funkce DTW v rovině (*i*,*j*), je nutné respektovat některé podmínky.

Okrajové podmínky:

$$i(1) = 1, \qquad j(1) = 1,$$

$$i(K) = I, \qquad j(K) = J. \tag{2.4}$$

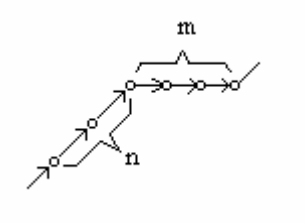
Podmínky spojitosti:

$$0 \le i(k) - i(k-1) \le Y,$$
$$0 \le j(k) - j(k-1) \le Z. \tag{2.5}$$

Okrajové podmínky definují, že první a poslední framy obou slov musí být vzájemně přiřazeny. Podmínky spojitosti definují vlastnosti (parametry) průběhu funkce. Funkce f musí být spojitá a monotónní. Průběh referenční posloupnosti lze modifikovat buď zopakováním framu s indexem *j*, použitím běžného framu nebo vynecháním framu s indexem *j*-1. Tyto podmínky limitují maximální a minimální strmost funkce.

Pro funkci DTW není vhodný ani příliš malý ani příliš velký přírůstek. Z těchto důvodů se bod $\mathbf{c}(k)$ nemůže pohybovat více jak m-krát v jednom směru za sebou, aniž by se předtím pohyboval v jiném směru n-krát (obr. 4). Velikost lokálního omezení strmosti může být ohodnocena mírou

$$p = n/m \tag{2.6}$$



Obr. 2.4 Znázornění lokálního omezení strmosti funkce DTW

Dalším omezením pro funkce je globální vymezení oblasti pohybu funkce DTW [1]. Kolísání v časové oblasti nemá nikdy za následek zásadní časové rozdíly mezi obrazy A a B. Lze proto stanovit, že

$$| i(k) - j(k) | \leq w, \tag{2.7}$$

kde w je vhodné celé kladné číslo. Zobecněním podmínky lokálního omezení strmosti funkce DTW na celou rovinu (i,j), lze vymezit přípustnou oblast průchodu funkce DTW rovinou (i,j).

Skutečnou minimální celkovou vzdálenost mezi obrazy A a B můžeme získat úpravou vztahu (2.3)

$$D(a,B) = \min_{c} \left[\frac{\sum_{k=1}^{K} d[\mathbf{c}(k)] q_c(k)}{Q(q_c)} \right], \tag{2.8}$$

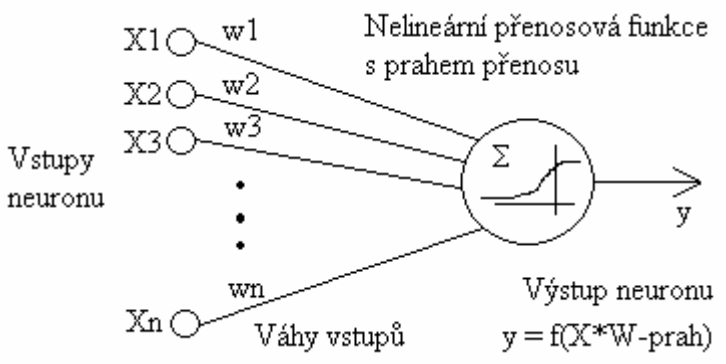
kde $d[\mathbf{c}(k)]$ je lokální vzdálenost mezi i-tým příznakem testovaného obrazu A a j-tým příznakem referenčního obrazu B v k-tém kroku, $q_c(k)$ je váhová funkce a $Q(q_c)$ značí normalizační faktor, který je funkcí váhové funkce.

Při rozpoznávání je reprezentace neznámého slova orientována podél osy i. Reference je orientována podél osy j. Vzdálenosti pro jednotlivé reference se počítají pro stejnou délku I. Tím je zajištěno, že vzdálenosti naměřené pro jednotlivé reference jsou porovnatelné. Poté se hledá reference, která je neznámému slovu nejblíže. Tato

metoda se však už moc nepoužívá, neboť ji vystřídala metoda založená na skrytých Markovových modelech.

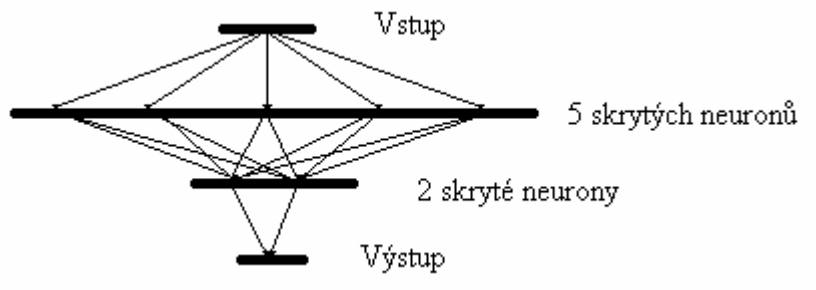
2.2 Neuronové sítě

Biologické neuronové sítě, ať už je jejich složitost jakákoliv, jsou velice výkonnými stroji v řešení nejrůznějších aplikací. Proto v polovině 80 dochází k vývoji umělých neuronových sítí (Artifical Neural Network – ANN). Neuronové sítě se převážně využívají pro rozpoznávání nejrůznějších signálů (obrazů). Umělá neuronová síť je distributivní, adaptivní, obecně nelineární stroj se schopností učení se. Je sestavena z mnoho prvků, které zpracovávají informace. Základním prvkem je model neuronu. Neuron je reprezentován hradlem, které plní určitou logickou funkci. Obecně může mít několik vstupů a několik výstupů. Počet se liší podle jeho zaměření (vstupní, výstupní neuron). Každý prvek je spojen s jinými prvky nebo také sám se sebou zpětnovazebním spojením. Informace mezi jednotlivými neurony se přenáší elektrickými impulsy. Tyto impulsy se v neuronu akumulují (zvyšuje se na něm napětí). Po překročení určitého prahu, se impuls vyšle z výstupů dál do sítě. Každý výstup má však jiný náboj. Po vstupu dalších neuronů do spojení, se náboj změní (výstup je zatížen).



Obr. 2.5 Matematický model neuronu

V dnešní době existuje řada typů neuronových sítí. Je důležité zvolit pro každý řešený úkol správnou architekturu sítě. Pro většinu aplikací je stále nejvíce preferovanou neuronovou sítí dopředná mnohovrstvá neuronová síť. Sítě mohou být několikavrstvé a mohou obsahovat vrstvy se skrytými neurony.



Obr. 2.6 Příklad neuronové sítě

Z hlediska průchodu informací se neuronovou sítí rozlišují na:

- *dopředné sítě* (feed forward) – tok informací prochází přímo od vstupu k výstupu sítě
- *rekurentní sítě* (reccurent) – informace prochází jak přímo tak i zpětnými vazbami na vstupy nebo do jiných vrstev

Z hlediska hodnot parametrů se neuronové sítě rozlišují na:

- *statické* – parametry sítě jsou nastaveny v procesu trénování, parametry jsou dále konstantní
- *dynamické* – parametry sítě jsou přednastaveny procesem trénování a dále se mohou měnit

Pro dobré fungování sítě je nutné správně nastavit váhové a prahové hodnoty. Algoritmus, který tyto hodnoty nastavuje se nazývá trénování (učení). Jsou dva druhy učení. Prvním je tzv. učení s učitelem a druhým je učení bez učitele. Trénování s učitelem je optimalizační algoritmus pro nalezení minima kritéria. Tento algoritmus nastavuje parametry sítě a srovnává výstup s kritériem tak dlouho, dokud nedosáhne požadované přesnosti. K tomu je potřeba vhodná tréninková skupina. Obrazy v tréninkové skupině (např.: obrazy obličejů nebo přímo oblasti zájmu) se nejprve vhodně upraví. Obrazy mohou být modifikovány hranovým detektorem, komprimovány pomocí PCA nebo DCT. Poté jsou použity pro trénování sítě. Po dostatečném natrénování lze síť použít k rozpoznávání.

2.3 Skryté Markovovy modely

Skrytý Markovův model (hidden Markov model – HMM) je matematický aparát, který je velmi podobný konečnému automatu. HMMs je jedna z metod, která se používá ve výpočetní technice k rozpoznávání a modelování přirozeného jazyka. HMMs se běžně využívají v těchto odvětvích: rozpoznávání a modelování lidské řeči, rozpoznávání ručně psaného písma, rozpoznávání lidského obličeje apod. Přístup k modelování lidské řeči pomocí skrytých Markovových modelů je založen na statistických metodách, ve kterých je promluva reprezentována jako pravděpodobnostní funkce. První pokus s tímto modelem provedl začátkem osmdesátých let Vintsyuk. Touto metodou se začala zabývat firma IBM a v roce 1985 představila kvalitní systém pro rozpoznání řeči – Tangora. Princip této metody vychází z představy o skutečném průběhu generování řeči. Hlasový přístroj se během krátkého časového úseku dostává do jednoho z konečného počtu stavů artikulačních konfigurací. V tomto časovém úseku sledujeme určitý parametr signálu tzv. spektrální charakteristiku. Je nutné, aby počet těchto charakteristik byl konečný. Tento problém řeší vektorová kvantizace. To znamená, že se vytvoří kódová kniha spektrálních vzorů. Každou analyzovanou spektrální charakteristiku je pak možno nahradit tím vzorem z kódové knihy, ke kterému je nejblíže. Při modelování řeči se využívá Markovova procesu. Generují se dvě navzájem časově závislé konečné posloupnosti náhodných proměnných. První je podpůrný Markovův řetězec, který je posloupností výše uvedených stavů a druhou je řetězec spektrálních vzorů. Zde se využívá konečného počtu spektrálních vzorů, které bereme z kódové knihy, a ke každému vzoru vytváříme náhodnostní funkci, která pravděpodobnostně ohodnocuje jeho vztah ke všem stavům. Předpokládá se, že v diskrétních časových okamžicích je tento proces v jediném stavu a tento stav lze identifikovat právě pomocí náhodnostní funkce, která odpovídá běžnému stavu. Podpůrný Markovovův řetězec pak mění své stavy podle matice pravděpodobnosti přechodu. Tyto stavy ale nejsou pro uživatele viditelné, jediné co lze zjistit, je výstup náhodnostních funkcí. Proto se tomuto modelu říká skrytý Markovovův model.

Skrytý Markovův model je ve své podstatě přímo konečným automatem. Jako automat má i skrytý Markovův model pevně zadanou množinu stavů, do kterých se může v průběhu výpočtu dostat, ale přechodová funkce je tvořena maticí přechodu a množina koncových stavů je nahrazena maticí pravděpodobnosti generovaných vzorů.

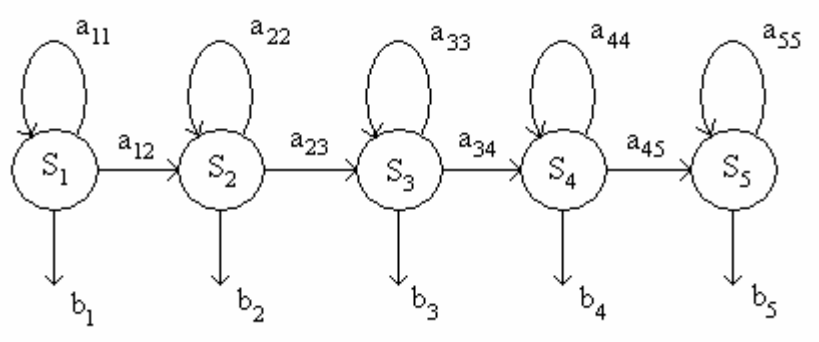
Je-li N počet stavů modelu $q_1, q_2, ..., q_N$, M počet spektrálních vzorů $o_1, o_2, ..., o_M$, lze definovat parametry Markovova modelu

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}), \tag{2.9}$$

kde $\boldsymbol{A} = [a_{ij}]$ je matice přechodu (určuje, s jakou pravděpodobností přechází systém ze stavu q_i v kterémkoliv čase t do stavu q_j v čase t+1), $\boldsymbol{B} = [b_{jm}]$ je matice pravděpodobnosti generovaných vzorů určuje, s jakou pravděpodobností je v kterémkoliv čase t generována m-tá položka konečného souboru spektrálních vzorů), $\boldsymbol{\pi} = [\pi_i]$ je sloupcový vektor pravděpodobnosti počátečního stavu. Platí tedy že,

$$\begin{aligned} a_{ij} &= P(q_{t+1} = q_j \mid q_t = q_i), && 1 \le i, j \le N, \\ b_{jm} &= b_j(m) = P(o_t = o_m \mid q_t = q_j), && 1 \le j \le N, 1 \le m \le M, \\ \pi_i &= P(q_1 = i), && 1 \le i \le N. \end{aligned} \tag{2.10}$$

Při modelování mluvené řeči se využívají tzv. levo-pravé Markovovy modely. Proces začíná příchodem prvního spektrálního vzoru z počátečného stavu modelu a s postupujícím časem setrvává ve stejném stavu nebo přechází do stavu dalšího. Výpočet se tedy provádí z leva doprava. Proces končí příchodem posledního spektrálního vzoru. Model se nachází v koncovém stavu. Původní typ Markovova modelu měl 40 až 50 stavů (tzv. Vintsyukův typ modelu). Během vývoje se ukázalo, že je počet stavů možno výrazně omezit a to na 4 až 7 stavů. Při tomto omezení nedošlo k výraznému poklesu přesnosti. Také trénování se stalo mnohem jednodušším. Toto omezení je možné díky tomu, že si jsou sousední stavy natolik podobné, že je můžeme sloučit bez ztracení přesnosti. V této práci bylo v testech použito jak pro dct příznaky tak i pro příznaky geometrické postupně 1 až 14 stavů.



Obr. 2.7 5stavovový skrytý Markovův model slova

2.3.1 Stanovení pravděpodobnosti promluvy

Úkolem modelu je pro libovolnou pozorovanou posloupnost vektorů příznaků ***O***, najít co nejpravděpodobnější posloupnost slov. Čili, že analyzovaná promluva $\boldsymbol{O} = o_1, o_2, ..., o_T$ byla generována modelem $\lambda = (\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$. Každé slovo *w*, resp. posloupnost slov *W*, je modelována odpovídajícím skrytým Markovovým modelem λ. Pravděpodobnost $P(\boldsymbol{O}| W)$ lze nahradit výpočtem pravděpodobnosti $P(\boldsymbol{O}| \lambda)$. Pro určení této pravděpodobnosti byl navržen efektivní způsob, tzv. algoritmus forvard-backward.

Výpočet odpředu (forward):

Nechť proměnná $\alpha_t(i)$ je pravděpodobnost generování částečné posloupnosti $(o_1, o_2, ..., o_t)$ a stavu $q_t = i$ při daném modelu λ. Platí

$$\alpha_t(i) = P(o_1, o_2, ..., o_t, q_t = 1 | \lambda) . \tag{2.11}$$

Hodnoty $\alpha_t(i)$ lze počítat rekurzivně podle následujícího algoritmu - *forward algorithm*:

1. Inicializace

$$\alpha_1(j) = \pi_j b_j(o_1), \qquad 1 \le j \le N \tag{2.12}$$

2. Rekurze

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij,} \qquad 1 \le j \le N, \qquad 1 \le t \le T-1 \tag{2.13}$$

3. Výsledná pravděpodobnost

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.14}$$

Výpočet odzadu (backward):

Nechť proměnná $\beta_t(i)$ je pravděpodobnost generování částečné posloupnosti $o_{t+1}, o_{t+2}, ..., o_T$ a stavu $q_t = i$ při daném modelu λ. Platí

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T \mid q_t = i, \lambda). \tag{2.15}$$

Hodnoty $\beta_t(i)$ lze opět počítat rekurzivně podle následujícího algoritmu - *backward algorithm:*

1. Inicializace

$$\beta_T(i) = 1, \qquad 1 \le i \le N \tag{2.16}$$

2. Rekurze

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \qquad 1 \le i \le N, \qquad 1 \le t \le T-1 \tag{2.17}$$

Pak platí

$$\alpha_t(i)\beta_t(i) = P(\boldsymbol{O}, q_t = i \mid \lambda), \qquad 1 \le i \le N, \qquad 1 \le t \le T \tag{2.18}$$

Použitím obou algoritmů lze určit $P(\boldsymbol{O}|\lambda)$ tak, že

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} P(\boldsymbol{O}, q_t = i \mid \lambda) = \sum_{i=i}^{N} \alpha_t(i)\beta_t(i) \tag{2.19}$$

Viterbiho algoritmus

Je to algoritmus, který řeší problém výpočtu pravděpodobnosti rekurzivně využitím techniky dynamického programování. Při výpočtu $P(O|\lambda)$ jsou ve výsledku zahrnuty pravděpodobnosti všech možných posloupností stavů délky T vzhledem k pozorované posloupnosti O. Pravděpodobnost generování posloupnosti O modelem λ lze alternativně nahradit výpočtem pravděpodobnosti optimální posloupnosti stavů za předpokladu posloupnosti O a modelu λ. Algoritmus hledá maximální pravděpodobné posloupnosti stavů. Definujme pomocnou veličinu jako maximální pravděpodobnost cesty DTW v čase *t*, která bere v úvahu prvních *t* pozorování výstupní posloupnosti $o_1, o_2, ..., o_{t-1}$ a končí v čase *t* ve stavu *i*.

1. Inicializace

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P(q_1, q_2, ..., q_{t-1}, q_t = i, o_1, o_2, ..., o_{t-1} \mid \lambda), \tag{2.20}$$

$$\delta_1(j) = \pi_j b_j(o_1), \qquad 1 \le j \le N \tag{2.21}$$

2. Rekurze

$$\delta_{t+1}(j) = b_j(o_{t+1})\left[\max_{1 \le i \le N} \delta_t(i) a_{ij}\right], \qquad 1 \le i \le N, \qquad 1 \le t \le T-1 \tag{2.22}$$

Procedura začne spočtením $\delta_T(j), 1 \le j \le N$ a použitím rekurze, přičemž je potřeba si zapamatovat, z kterého stavu v předchozím kroku byla vybrána maximální hodnota, a to v každém časovém kroku *t*. Výsledný index maximálně pravděpodobného stavu j* v čase *t*=*T* je roven

$$j^* = \arg\max_{1 \le j \le N} \delta_T(j) \tag{2.23}$$

Pro *t*=*T*-1,*T*-2,…,1 lze indexy hledaných stavů určit zpětným trasováním stavů s největší pravděpodobností.

2.3.2 Trénování parametrů modelu

Při používání metody skrytých Markovových modelů je nejdůležitější správné natrénování modelů. V úloze rozpoznávání izolovaných slov je rozpoznávaná promluva vždy jen jedno slovo. Pro jedno slovo je natrénován jeden Markovův model. To znamená nastavit jeho parametry λ tak, aby byla tímto modelem maximalizována pravděpodobnost určení promluvy. Je snahou, aby tato pravděpodobnost byla globálním maximem. Zatím se však tento problém nedaří analyticky vyřešit. Tuto pravděpodobnost je možné maximalizovat pouze lokálně. K tomu byly navrhnuty určité iterativní postupy. Nejznámějším a nejpoužívanějším postupem je Baumův-Welchův algoritmus. Pracuje na principu cyklického opakování odhadu, pro který lze na základě trénovací množiny a stávajícího modelu určit nový odhad parametrů λ. To znamená, že se iterativně aktualizují parametry Markovova modelu. Definuje se proměnná $\xi_t(i, j)$, která vyjadřuje pravděpodobnost, že proces je v čase *t* ve stavu *i* a v čase *t*+1 ve stavu *j* za předpokladu pozorované posloupnosti ***O*** a daného modelu λ

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j \mid \boldsymbol{O}, \lambda)$$

neboli $$\xi_t(i,j) = \frac{P(q_t = i, q_{t+1} = j, O \mid \lambda}{P(O \mid \lambda)} \tag{2.24}$$

Z předchozího forward-backward algoritmu lze odvodit

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^{N} \sum_{j'1}^{N} \alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})} \tag{2.25}$$

Dále je definována proměnná $\gamma_t(i)$, která určuje pravděpodobnost, že model je v čase *t* ve stavu *i* při dané pozorované posloupnosti ***O*** a daného modelu λ. Z forward-backward dostaneme

$$\gamma_t(i) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \right] \tag{2.26}$$

Platí

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \qquad 1 \le i \le N, \qquad 1 \le t \le M \tag{2.27}$$

Provedením sumace $\gamma_t(i)$ podle času (od *t*=1 do *t*=*T*-1), se získá očekávaný počet přenosů realizovaných ze stavu *i*. Obdobně suma pro $\xi_t(i,j)$ udává očekávaný počet přenosů ze stavu *i* do stavu *j*. Pro model $\boldsymbol{\lambda} = \boldsymbol{(A,B,\pi)}$, podle předchozího se určí α, β, ξ a ξ. V dalším kroku se odhadují nové parametry, tzv. reestimace parametrů:

$$\boldsymbol{\pi}_i = \gamma_1(i), \qquad 1 \le i \le N \tag{2.28}$$

$$\mathbf{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \xi_t(i)}, \qquad 1 \le i \le N, \qquad 1 \le j \le N \tag{2.29}$$

$$\mathbf{b}_j(m) = \frac{\sum_{t=1}^{T} \xi_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \qquad 1 \le j \le N, \qquad 1 \le m \le M \tag{2.30}$$

Reestimační vztahy pro odhady nových parametrů tedy iterativně maximalizují pravděpodobnost $P(\boldsymbol{O}|\ \lambda)$. Je možné je odvodit i jiným způsobem, například využitím Lagrangeovy metody neurčitých koeficientů.

2.3.3 Rozhodovací kritérium

Pro neznámé slovo, které je reprezentováno analyzovanou posloupností spektrálních vzorů $O = o_1, o_2, ..., o_T$, se určují pravděpodobnosti, s jakými by danou neznámou posloupnost generovaly jednotlivé Markovovy modely. Neznámé slovo je pak zařazeno do takové třídy $L_m^l,\ 1 \le m \le N,$ pro kterou byla určená pravděpodobnost maximální, tedy

$$l^* = \arg \max_{1 \le m \le N} L_m^l. \tag{2.31}$$

1.1 Výpočet všech pravděpodobností se provádí výše popsaným algoritmem forward-bacward.

2.3.4 HTK Toolkit

HTK Toolkit [2] je soubor programů pro práci se skrytými Markovovými modely. HTK se skládá z konzolových aplikací. Konzolové aplikace se hodí pro automatizované úlohy, kterých je v úloze rozpoznávání většina. Všechny programy jsou programy spustitelné z příkazové řádky s parametry. Toho lze dobře využít při tvorbě komplexních úloh, kde se využívají dávkové soubory. Pomocí parametrů se programům určují např. vstupní data, výstupní adresáře, nastavení apod. V této práci byly používány převážně tyto programy: hinit, hrest a hvite. Hinit se používá pro počáteční odhadnutí parametrů HMMs. Hrest vykonává algoritmus základní Baumovi-Welchovi reestimace pro nalezení parametrů HMMs. Hvite je univerzální Viterbiho rozpoznávač slov.

3. Experimenty

Při procesu rozpoznávání řeči je nutné si nejprve připravit dostatečně velkou knihovnu trénovacích dat (například soubor jednotlivých izolovaných slov). K trénování a zároveň rozpoznávání byla použita data z audiovizuální databáze AVDbcz1. Jednotlivé promluvy byly vhodně parametrizovány a to pomocí příznaků získaných z DCT (n05-047dct.par) a geometrických příznaků (n05-047g.par). Každý par soubor obsahuje vektory příznaků. A to buď 14 dct příznaků nebo 14 příznaků geometrických. Hledá se nejvhodnější vektor příznaků pro dosažení co největší pravděpodobnosti správného rozpoznání testovaného slova. Používáme tedy vždy jen určitou část těchto příznaků. Tyto příznaky se musí z původního par souboru vykopírovat a vytvořit nový par soubor stejného jména, ale už jen s určitými příznaky. Tato část problému byla řešena programem ParModifier. Pomocí něho se vybírají jednotlivé příznaky a vytváří se nové modifikované par soubory. Cesta k jednotlivým původním par souborům je ukládána do textových souborů, které se vytváří programem ParLinker. V audiovizuální databázi je 50 slov od 30 mluvčích, což je 1500 souborů. Pro zpracovávání takto velkého počtu souborů byly použity dávkové soubory (např.: mybatdct.bat), které zajišťují nepřetržitý tok vstupních (trénovacích) dat pro programy, které je zpracovávají. Nové par soubory jsou dále zpracovávány MakeProtoHMMSet.PL skriptem, který vytváří prototypy jednotlivých slov. Informace o souboru (parametry každého par souboru) jsou čteny ze souboru proto.pcf. V něm jsou uloženy všechny důležité údaje, jako je počet příznaků (vecsize), počet stavů (nStates) apod. Tento soubor je upravován pomocí programu ProtoChanger. Dále jsou prototypy inicializovány voláním init.bat, respektive voláním programu hinit.exe z HTK Toolkitu. Inicializované soubory pak trénovány. To znamená, že jsou vytvářeny modely jednotlivých izolovaných slov. Pro trénování je určen program hrest. Hned po natrénování slov následuje jejich rozpoznávání. Rozpoznávanými slovy je 50 slov od 5 mluvčích. Jednotlivá slova jsou rozpoznávána programem hvite. Jeho výstupem jsou soubory roz1.txt až roz.50.txt, ve kterých jsou uloženy indexy slov nejpravděpodobněji odpovídajících rozpoznávanému slovu. Tyto soubory jsou čteny programem DataPickuper, který tyto informace vykopírovává a ukládá je do snáze čitelného souboru DataPickUper.ini. Finálním výstupem celého testu je celkový přehled, kde ke každému vektoru příznaků a počtu stavů je přirazena pravděpodobnost úspěchu rozpoznání, která je vztažena na všech 50 slov od 30 mluvčích.

3.1 Dávkové soubory

Dávkové soubory, někdy též dávkové programy nebo skripty, se velice dobře osvědčily při vykonávání velkého počtu stejných operací nebo úloh. Dávkový soubor je neformátovaný textový soubor, do kterého jsou zapisovány příkazy. Může také obsahovat příkaz pro spuštění jiného dávkového souboru, které je nutné volat pomocí *call*. Příkazem *Call* je zaručeno vrácení se z vnořeného bat souboru a pokračování v provádění příkazů hlavního bat souboru. Tyto soubory mají příponu .BAT nebo .CMD. Jsou to spustitelné soubory, které zpracovává příkazový procesor cmd.exe. Ten vykonává jednotlivé příkazy postupně od shora dolů tak, jak jsou zapsány v souboru. Aby bylo možné volat dávkové a spustitelné programy z příkazové řádky, musí být tyto soubory uloženy v proměnném prostředí. Proměnné prostředí je pojmenovaný objekt, který uchovává nějakou informaci používanou jedno nebo více aplikacemi. Toto prostředí lze specifikovat v nastavení operačního systému.

3.1.1 Použité dávkové soubory

Příkazy pro celý jeden kompletní test obsahuje dávkový soubor mybatdct.bat. V něm je uložena posloupnost příkazů pro celý průběh testu. Vše od modifikace par souborů až po výstup s výsledky. V příkazové řádce je spuštěn jediný bat soubor a to mybatdct.bat. Ten obsahuje odkazy na další dávkové soubory: Parset.bat a ProtoSet.bat. V ParSet.bat je spouštěn ParModifier.exe, který modifikuje (vytváří nové) par soubory určené ke zpracování. ProtoSet.bat obsahuje posloupnost příkazů, které vedou k inicializování, trénování a následnému rozpoznávání slov.

```
ParLinker.exe
call ParSet1.bat
call ProtoSet1.bat
…
call ParSet7.bat
call ProtoSet7.bat
PercentageWiever.exe m
```

Zdr. kód 3.1 Část příkazů hlavního dávkového souboru mybatdct.bat.

```
@echo off
echo.
@echo ParModifier.exe is working...
@echo on
ParModifier.exe ParLinks.txt 1,2,3,4,5
@echo off
echo.
@echo All PAR files were modified.
@echo on
```

Zdr. kód 3.2 Úplný sled příkazů vnořeného ParSet1.bat souboru.

```
ProtoChanger.exe proto.pcf 1 5
MakeProtoHMMSet.PL proto.pcf
call init.bat
call train.bat
call roz.bat
DataPickUper.exe m
…
ProtoChanger.exe proto.pcf 14 5
MakeProtoHMMSet.PL proto.pcf
call init.bat
call train.bat
call roz.bat
DataPickUper.exe m
```

Zdr. kód 3.3 Část příkazů vnořeného ProtoSet1.bat souboru.

V bat souborech init.bat, train.bat a roz.bat jsou spouštěny přímo programy HTK Toolkitu, který je určen pro rozpoznávání pomocí HMM. Jsou psány podle syntaxických pravidel vyžadujících všechny HTK programy.

```
hinit -S t01.txt -C config.txt -M inited prototype/slovo01
…
hinit -S t50.txt -C config.txt -M inited prototype/slovo50
```

Zdr. kód 3.4 Část příkazů vnořeného init.bat souboru.

```
hrest -S t01.txt -C config.txt -i 100 -M train inited/slovo01
…
hrest -S t50.txt -C config.txt -i 100 -M train inited/slovo50
```

Zdr. kód 3.5 Část příkazů vnořeného train.bat souboru.

```
hvite -i roz01.txt -w sit/wdnet -S seznam01.txt -n 20 10 -C config.txt
-H train/slovo01 … -H train/slovo50
…
hvite -i roz50.txt -w sit/wdnet -S seznam50.txt -n 20 10 -C config.txt
-H train/slovo01 … -H train/slovo50
```

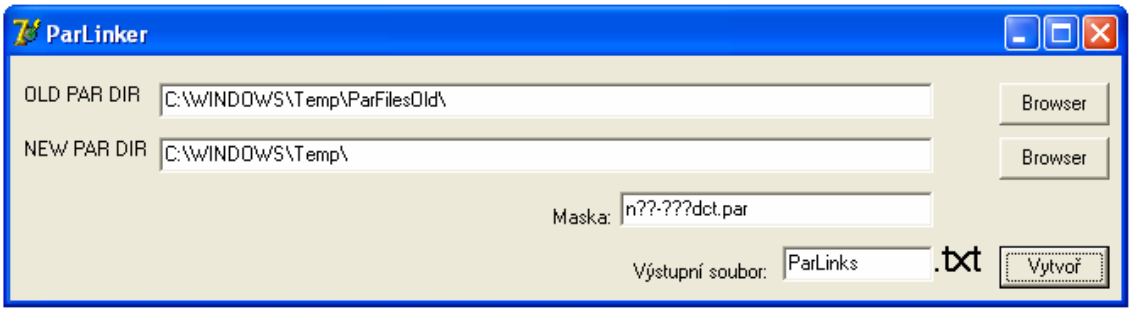
Zdr. kód 3.6 Část příkazů vnořeného roz.bat souboru.

3.2 Použité programy

Celý proces testování vyžaduje řešení mnoha různých úloh. Pro snadnou práci bylo vytvořeno několik programů, které tyto úlohy vykonávají. Každý tedy řeší jen část problému. Všechny programy pracují ve dvou módech. První mód: program je spuštěn dvojklikem. V tomto módu lze využít veškeré funkce programu skrze uživatelsky příjemnější grafické rozhraní. Druhý mód: program je spuštěn voláním v bat soboru, kde se za jeho názvem nacházejí parametry (nastavení), s jakými se má daný program spustit. Jde tedy o volání programu s parametrem. Při programování byly použity dvě funkce. První vrací počet parametrů napsaných za názvem programu. Druhá vrací parametr, který koresponduje s indexem parametru.

3.2.1 ParLinker

Program ParLinker je určen k vytváření textových souborů obsahujících cesty (linky) na všechny par soubory, které potřebujeme zpracovat. Pomocí programu Parlinker se určuje: odkud se data budou brát, jaká data (maska: n??-???dct.par) a kam se budou ukládat. Program dokáže najít všechny par soubory obsažené v zadané složce. Bylo použito par souborů, které obsahovaly všechny druhy příznaků (statické, dynamické a akcelerační). Proto byl vytvořen jen jediný soubor s cestami. Tedy soubor, který obsahuje cesty ke všem 50 slovům od 30 mluvčích. Dále už není potřeba tyto cesty měnit, a proto se ParLinker volá hned na začátku dávkového souboru.



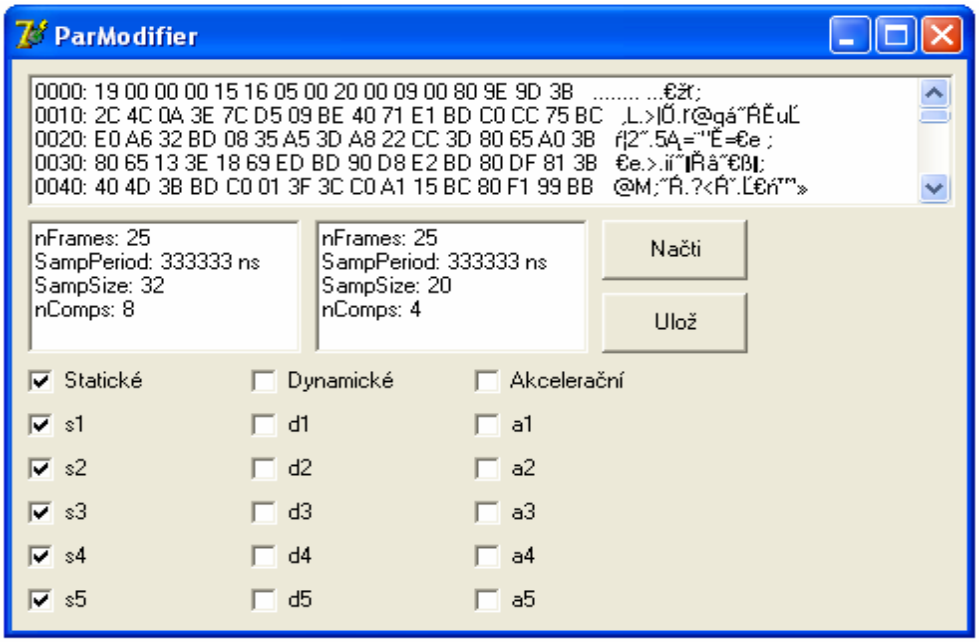
Obr. 3.1 Grafické rozhraní programu ParLinker.

```
C:\WINDOWS\Temp\ParFilesOldDct\n01-001dct.par
C:\WINDOWS\Temp\n01-001dct.par
...
C:\WINDOWS\Temp\ParFilesOldDct\n30-050dct.par
C:\WINDOWS\Temp\n30-050dct.par
```

Zdr. kód 3.7 Struktura linků ve výstupním souboru Padlinka.txt.

3.2.2 ParModifier

V prováděných testech byla hledána skupina příznaků a počet stavů, které by měly nejlepší rozpoznávací skóre. Proto byl počet a druh použitých příznaků postupně měněn. Z původních par souborů jsou vykopírovávány jen ty příznaky, které jsou právě používány a vytváří se nový par soubor obsahující právě tyto příznaky. Cesta k jednotlivým původním par soborům je čtena z ParLinks.txt, který byl vytvořen pomocí programu ParLinker. Příznaky v souboru mají pevně danou polohu. Jsou seřazeny hned za hlavičkou souboru postupně za sebou. Nejdříve 5 statický, pak 5 dynamických a nakonec 5 akceleračních. Při práci v grafickém módu je vypisován celý obsah souboru v hexa kódu, čtena data z hlavičky (nFrames, SampPeriod, SampSize, nComps) a lze zaškrtnout, jaké příznaky se mají vykopírovat. Při spouštění v příkazové řádce jsou vyžadovány tyto parametry: 1. název souboru obsahujícího cesty k původním par souborům. 2. číselné indexy příznaků, které chceme vykopírovat oddělené čárkou. Např.: `ParModifier.exe ParLinks.txt 1,2,3,4,5`



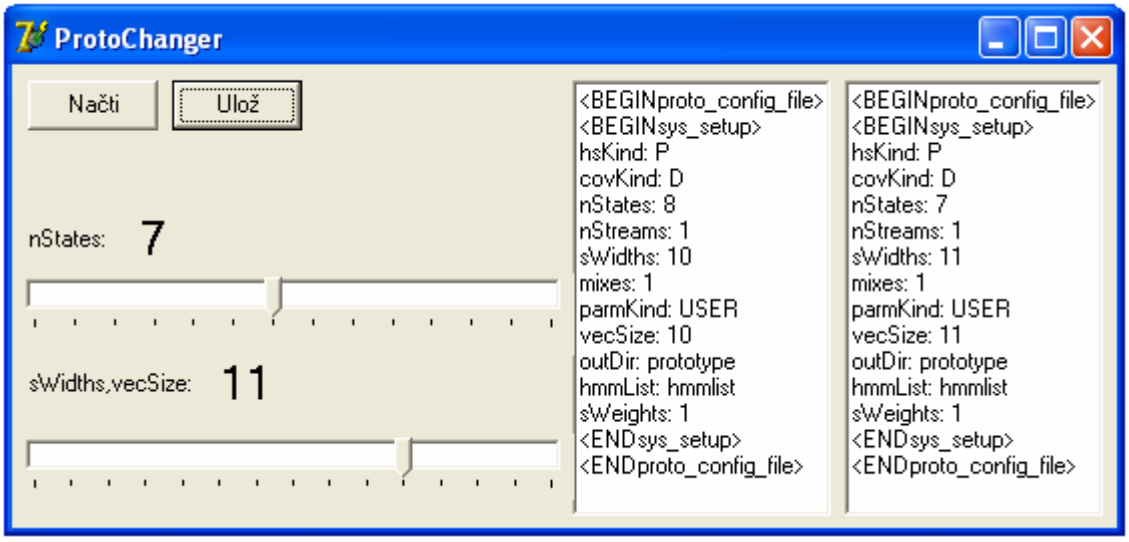
Obr. 3.2 Grafické rozhrání programu ParModifier.

3.2.3 ProtoChanger

Jak už bylo zmíněno výše, programem ProtoChanger jsou měněna data v souboru proto.pcf, který je zdrojem informací (nastavení) pro MakeProtoHMMSet.PL skript. Nejdůležitějšími údaji jsou nStates, což je počet stavů, a vecSize, který udává počet příznaků. Při spouštění v příkazové řádce jsou vyžadovány tyto parametry: 1. název souboru obsahujícího nastavení (v našich testech vždy proto.pcf). 2. počet stavů (nStates) 3. počet příznaků (vecSize). Např.: `ProtoChanger.exe proto.pcf 1 5`

```
<BEGINproto_config_file>
<BEGINsys_setup>
hsKind: P
covKind: D
nStates: 14
nStreams: 1
sWidths: 12
mixes: 1
parmKind: USER
vecSize: 12
outDir: prototype
hmmList: hmmlist
sWeights: 1
<ENDsys_setup>
<ENDproto_config_file>
```

Zdr. kód 3.8 Úplný obsah souboru proto.pcf.



Obr. 3.3 Grafické rozhrání programu ProtoChanger.

3.2.4 DataPickUper

Výsledky vypočítané pomocí hvite jsou ukládány do roz01.txt až roz50.txt souborů. V souborech je ke každému mluvčímu (jedno slovo od 5 mluvčích) vygenerována posloupnost deseti slov s nejvyšší pravděpodobností od nejvyšší po nejnižší, že toto slovo odpovídá slovu rozpoznávanému. Ve zdrojovém kódu (3.9) bylo 19. slovo rozpoznáno jako 36.

```
#!MLF!#
"C:/WINDOWS/Temp/n31-019g.rec"
0 7999992 s36 513.955872
///
0 7999992 s19 494.271881
///
0 7999992 s34 468.287292
///
0 7999992 s30 466.278229
///
0 7999992 s17 465.530640
///
0 7999992 s46 452.378510
///
0 7999992 s49 446.299622
///
0 7999992 s44 445.792023
///
```

```
0 7999992 s31 443.307800
///
0 7999992 s38 442.614624
```

Zdr. kód 3.9 Pravděpodobnosti pro 1. mluvčího 19. slovo (n31..35 je index mluvčích pro rozpoznávání).

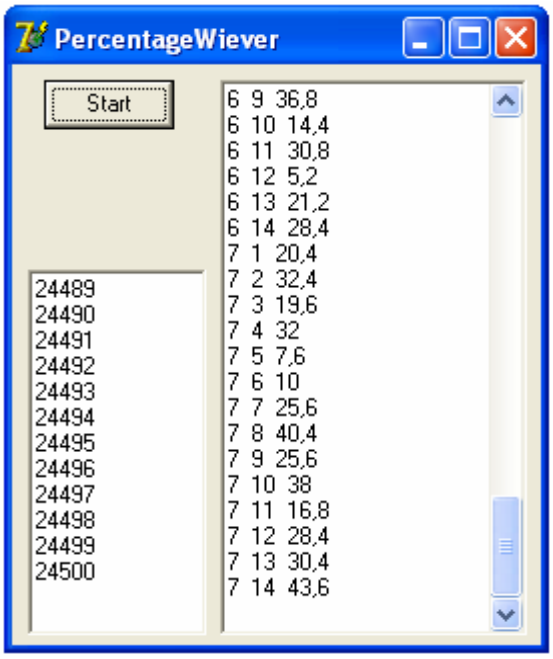
Tyto soubory jsou čteny a je z nich zjiš´továno, na jakém místě se umístilo námi rozpoznávané slovo. Tyto údaje jsou ukládány do DataPickUper.ini souboru pro další zpracování.

```
[roz38.txt64]
31=2
32=1
33=2
34=1
35=1
```

Zdr. kód 3.10 Část souboru DatapickUper.ini. Konkrétně výsledky rozpoznání 38. slova v 64. testu.

3.2.5 PercentageWiever

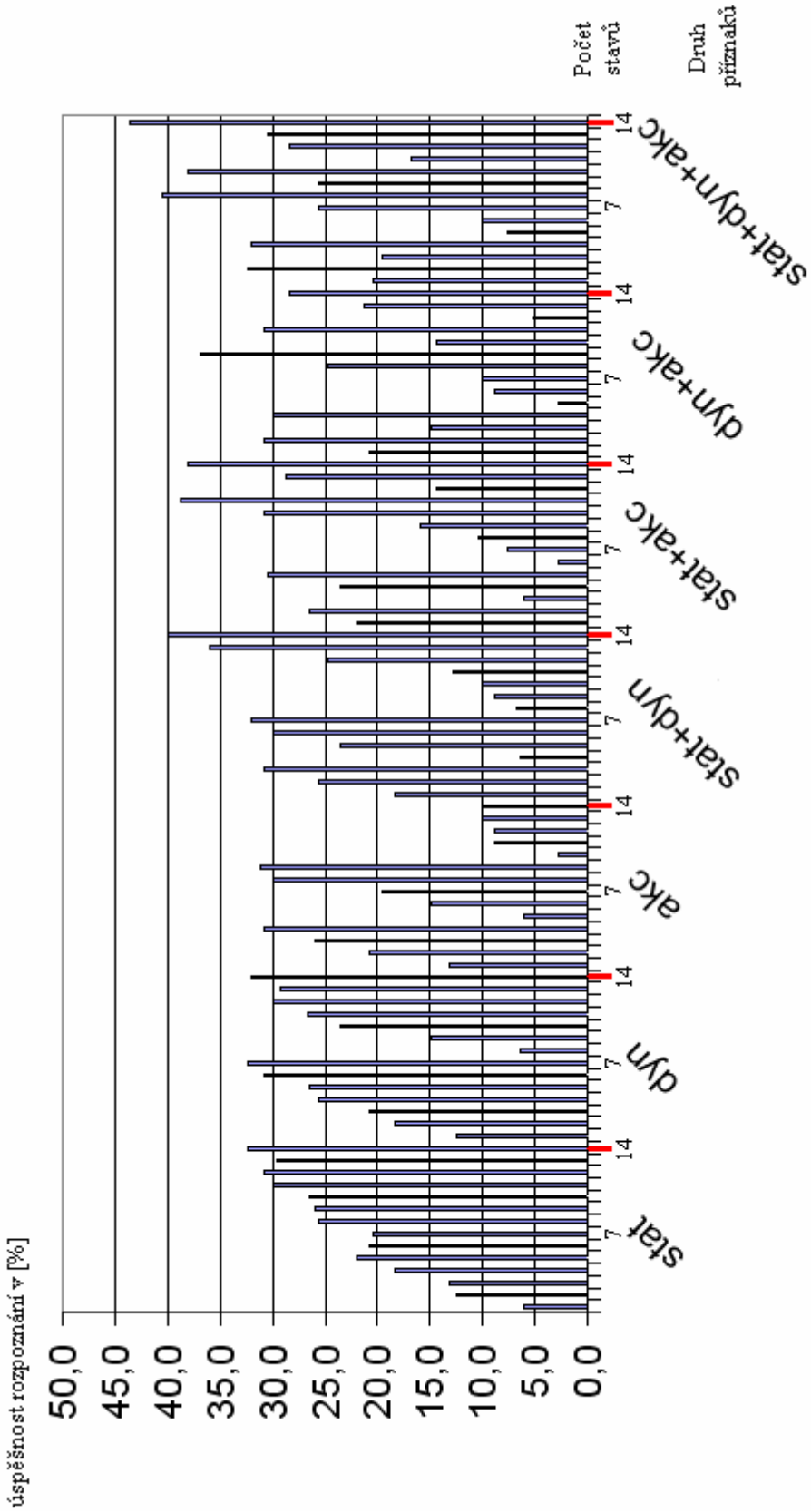
Posledním programem je PercentageWiever. Tím je zpracováván soubor DataPickUper.ini. Informace v něm obsažené jsou převáděny do ucelené a přehledné formy pro čtení. První číslo nalevo znamená druh příznaků (1 jsou statické, 2 dynamické,…, 5 statické+akcelerační,…). Druhé číslo reprezentuje počet stavů. Posledním je výsledná pravděpodobnost rozpoznání slov pro jednotlivá nastavení.



Obr. 3.4 Grafické rozhrání programu PercentageWiever.

3.3 Výsledky

Prvními dosaženými výsledky z testů byla procentuelní úspěšnost rozpoznání neznámého slova, které bylo reprezentováno DCT příznaky viz. graf (3.1), příloha č. 1.

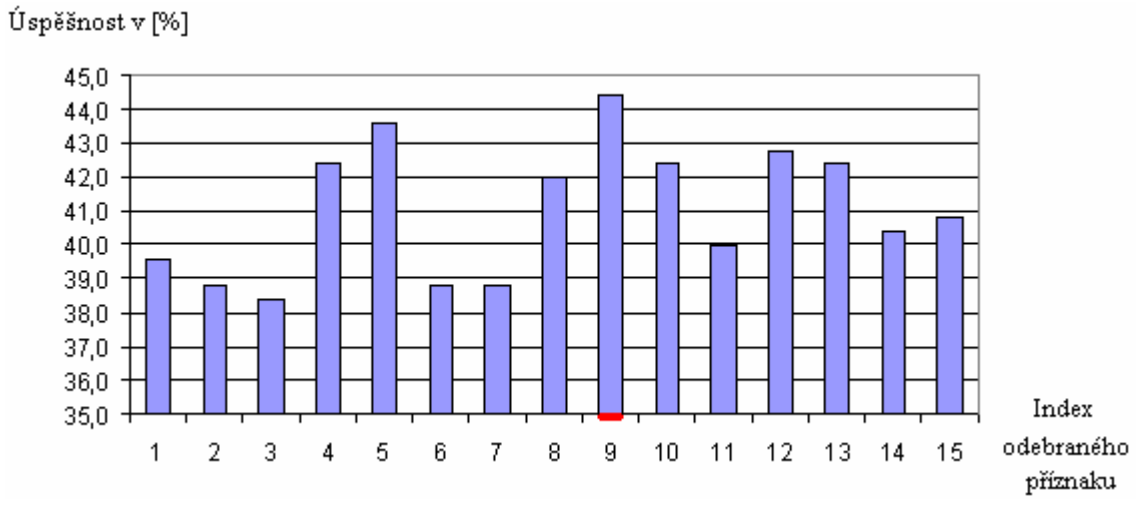


Graf 3.1 Dosažená úspěšnost pro DCT příznaky v [%] v závislosti na druhu příznaků a počtu stavů, zdrojová data jsou v tabulce č. 1 , která je součástí přílohy

Největší úspěšnost byla zaznamenána při testování slov, která byla reprezentována všemi příznaky (statické + dynamické + akcelerační) a HM model měl čtrnáct stavů, a to 43,6 %. V dalším testu byla snaha tuto pravděpodobnost zvýšit. Základní myšlenka byla, že některý z patnácti příznaků negativně ovlivňuje výslednou pravděpodobnost. Proto bylo provedeno čtrnáct dalších testů. V každém z nich byl odebrán vždy jen jeden příznak. V jednom případě se skutečně podařilo úspěšnost nepatrně zvýšit a to při odebrání čtvrtého dynamického příznaku (index 9) o 0,8 % na 44,4 %. Výsledky jsou zobrazeny v tabulce (3.1) a grafu (3.2).

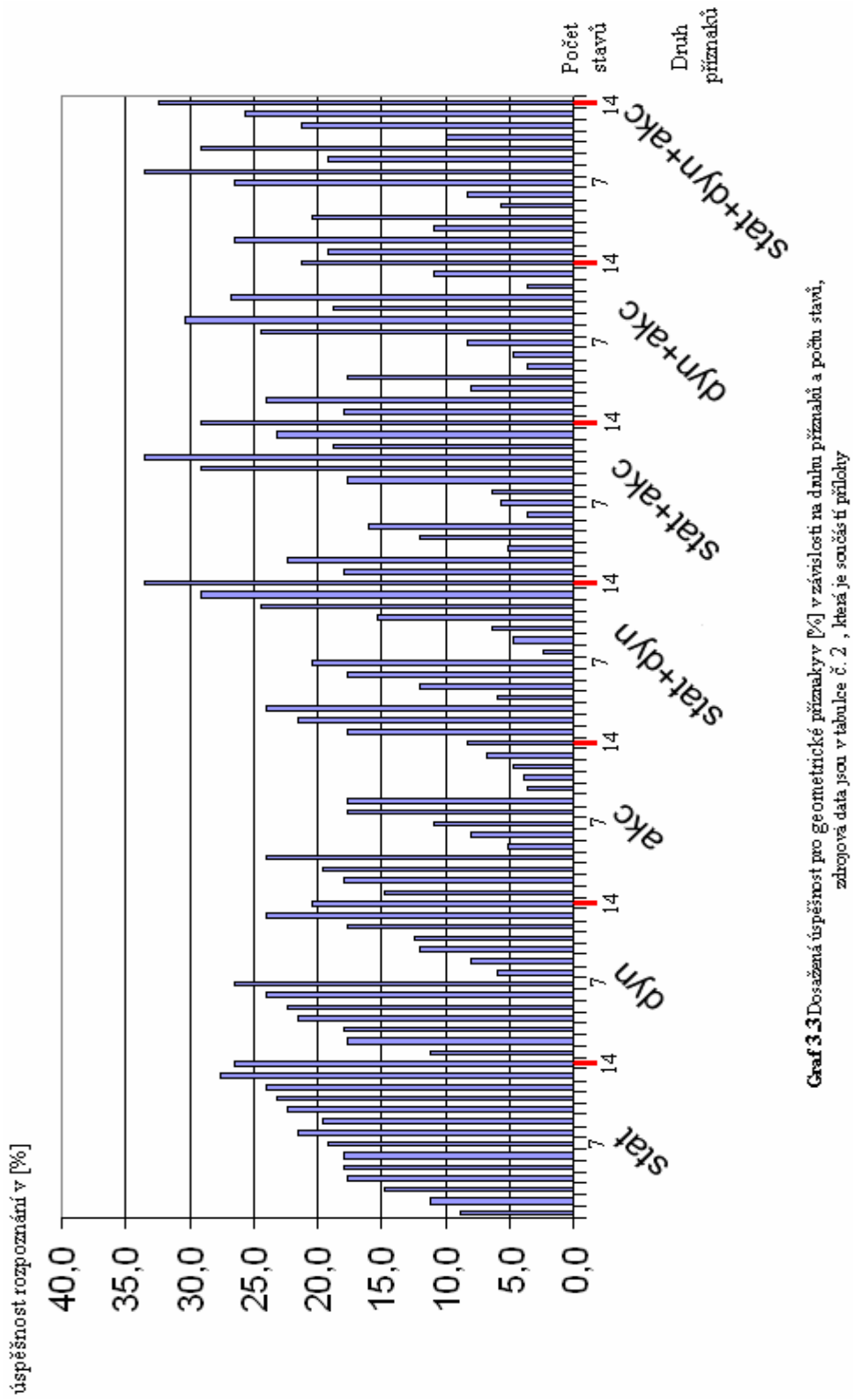
Index odebraného příznaku	Úspěšnost v [%]
1	39,6
2	38,8
3	38,4
4	42,4
5	43,6
6	38,8
7	38,8
8	42,0
9	44,4
10	42,4
11	40,0
12	42,8
13	42,4
14	40,4
15	40,8

Tab. 3.1 Úspěšnost rozpoznání závislá na odebraném příznaku v [%]



Graf 3.2 Úspěšnost rozpoznání závislá na odebraném příznaku v [%]

V dalším testu se pracovalo s příznaky geometrickými a byly opět zjištěny pravděpodobnosti úspěchu rozpoznání, viz. graf (3.3), příloha č. 1. Jako v prvním testu s DCT příznaky, tak i v tomto testu byly zkoušeny postupně všechny kombinace druhů příznaků (stat, dyn, akc, stat + dyn,…).



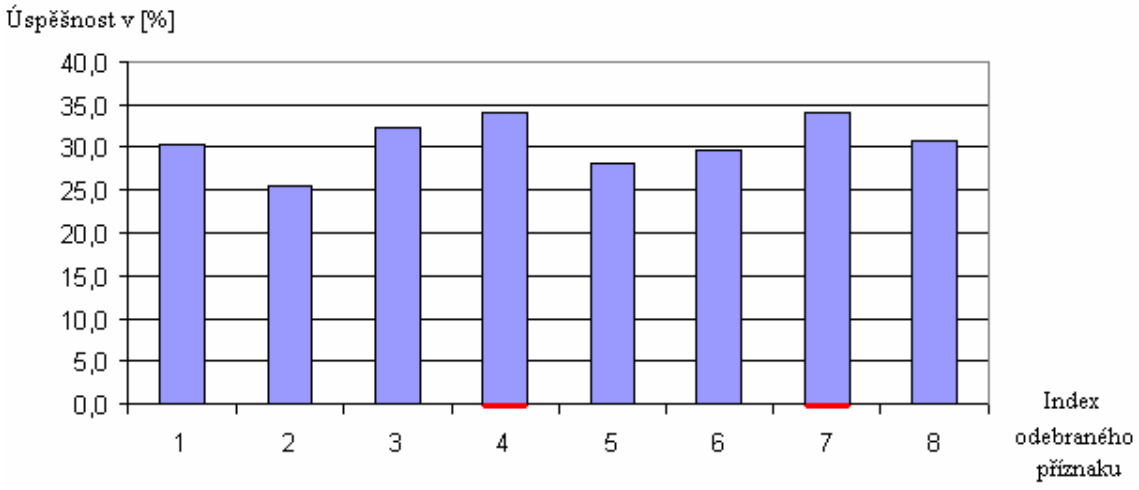
Graf 3.3 Dosažená úspěšnost pro geometrické příznaky v [%] v závislosti na druhu příznaků a počtu stavů, zdrojová data jsou v tabulce č. 2 , která je součástí přílohy

V tomto testu se podařilo dosáhnout vysokých úspěšností u dvou případů nastavení a to při rozpoznávání slov reprezentovaných statickými + dynamickými příznaky – 14 stavů, statickými + akceleračními – 11 stavů. U obou těchto případů dosáhla pravděpodobnost rozpoznání 33,6 %. Dále se zkoušelo zvýšit tyto pravděpodobnosti opět odebíráním jednoho z příznaků, viz tab. (3.2), graf (3.4) a (3.5).

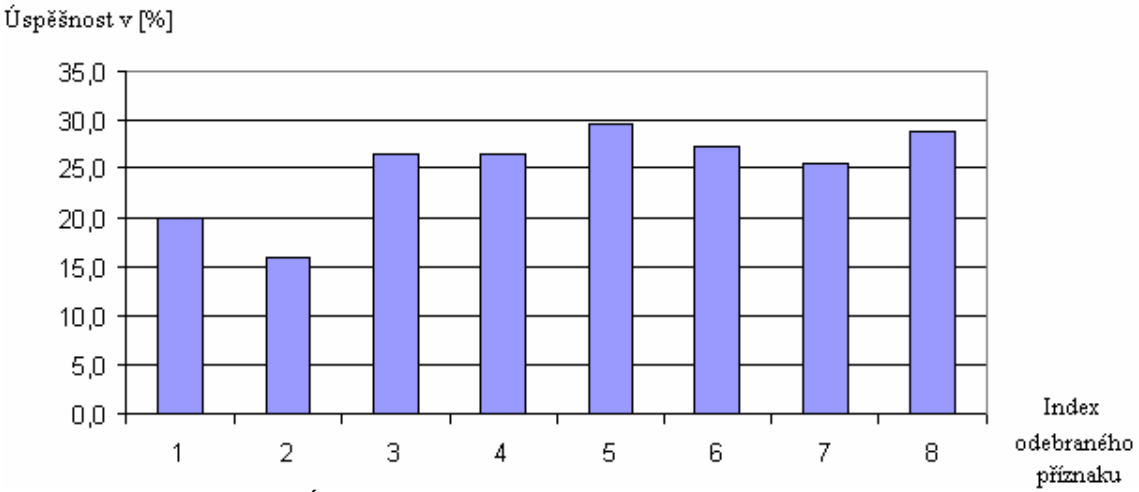
stat+dyn	1	30,4
stat+dyn	2	25,6
stat+dyn	3	32,4
stat+dyn	4	34,0
stat+dyn	5	28,0
stat+dyn	6	29,6
stat+dyn	7	34,0
stat+dyn	8	30,8

stat+akc	1	20,0
stat+akc	2	16,0
stat+akc	3	26,4
stat+akc	4	26,4
stat+akc	5	29,6
stat+akc	6	27,2
stat+akc	7	25,6
stat+akc	8	28,7

Tab. 3.2 Úspěšnost rozpoznání závislá na odebraném příznaku v [%]



Graf 3.4 Úspěšnost rozpoznání závislá na odebraném příznaku v [%]



Graf 3.5 Úspěšnost rozpoznání závislá na odebraném příznaku v [%]

Ke zlepšení pravděpodobnosti došlo pouze u příznaků statických + dynamických z 33,6% na 34,0 %. Odebírání příznaků u statických a akceleračních příznaků vedlo jen k poklesu pravděpodobnosti.

3.3.1 Shrnutí

Při rozpoznávání izolovaných slov reprezentovanými DCT vizuálními příznaky bylo dosaženo úspěšnosti správného rozpoznání 43.6 % a to při použití všech druhů příznaků (statické + dynamické + akcelerační) a čtrnácti stavů. Tuto pravděpodobnost se podařilo nepatrně zvýšit o 0,8 % na 44,4 % odebráním čtvrtého dynamického příznaku (index 9). V dalších testech byly používány příznaky geometrické. Při těchto testech bylo dosaženo maximální úspěšnosti 33,6 % hned ve dvou případech (příznaky statické + dynamické – čtrnáct stavů, statické + akcelerační – čtrnáct stavů). Zlepšení bylo dosaženo jen u příznaků statických + dynamických o 0,4 % na 34,0 %.

Závěr

V rámci této bakalářské práce byla upravena audiovizuální databáze AVDBcz1. Ze sekvence video snímků byly vytvořeny parametrizované soubory (např.: n01-001dct.par). Tyto soubory byly následně trénovány a rozpoznávány pomocí HTK Toolkitu. Pro úlohu parametrizování a rozpoznávání bylo vytvořeno několik programů.

Každé slovo je rozděleno do několika snímků. Na začátku a konci se nacházejí snímky, které neobsahují žádnou informaci o vyřčeném slově. Pro získání příznaků z vizuálního signálu a dosažení dobrého rozpoznávacího skóre je nutné mít zpracovávané slovo bez nadbytečných snímků (ořezané). To bylo provedeno pomocí program, kterým byl zobrazován průběh akustického signálu v čase. Slovo bylo ručně ořezáno. Zde se naskytl menší problém v podobě neschopnosti oříznout signál na první pokus. Na konci slova se mohou vyskytovat znělé hlásky, které mají neostré ukončení. Bylo nutné si vyříznutou část opět přehrát a zkontrolovat přesnost oříznutí. Znalost začátku a konce zvukového signálu byla použita při výběru video snímků pro parametrizaci. Dalším problémem byly samotné snímky. Ukázalo se, že obličej tvoří z velikosti snímku příliš malou část. Výsledkem parametrizace byly dva druhy parametrizovaných souborů. Jedny byly vytvořeny z DCT příznaků a druhé z příznaků geometrických.

S těmito soubory bylo provedeno několik testů. Postupně byly prostřídány všechny kombinace statických, dynamických a akceleračních příznaků a počet stavů, který se měnil od jedné do čtrnácti. Při použití DCT příznaků bylo dosáhnuto nejvyššího skóre 43,6 % a to při použití všech druhů příznaků (statické + dynamické + akcelerační), počet stavů byl čtrnáct. Byla také vyzkoušena myšlenka, že některý z příznaků může tuto pravděpodobnost negativně ovlivňovat. Proto byl postupně odebírán jeden příznak a slovo bylo znovu rozpoznáváno. Ke zlepšení úspěšnosti došlo jen v jednom případě. Úspěšnost se nepatrně zvýšila o 0,8 % na 44,4 % při odebrání čtvrtého dynamického příznaku. Při použití geometrických příznaků bylo dosaženo nejvyšší úspěšnosti 33,6 % ve dvou případech (statické + dynamické-14 stavů a statické + dynamické-11 stavů). Zlepšit pravděpodobnost rozpoznání se podařilo jen v prvním případě o 0,4 % na 34,0 %.

Rozpoznávání izolovaných slov nebo celých promluv je v dnešní době výrazně se rozvíjejícím oborem. Je to i díky rozvoji výpočetní techniky, která musí zpracovávat velká množství dat. Rozpoznávání mluvené informace z vizuálního signálu řeči se

používá pro podporu akustické informace. Toho by šlo využít při komunikaci s počítačem. Člověk sedí většinou čelně k monitoru. Na něm by mohla být umístěna malá kamera, která by snímala obličej. Na tuto práci lze tedy navázat spojením akustického a vizuálního signálu při rozpoznávání.

Literatura

[1] Zdeněk Kotek, Vladimír Mařík a kol.: *Metody rozpoznávání a jejich aplikace*, Academia 1993, Praha

[2] Josef Psutka, Luděk Miller, Jindřich Matoušek, Vlasta Radová: *Mluvíme s počítačem česky*, Academia 2006, Praha

[3] Technická unuverzita v Liberci: *Počítačové zpracování řeči*, TUL 2001, Liberec

[4] Josef Psutka: *Komunikace s počítačem mluvenou řečí*, Academia 1995, Praha

[5] Josef Chaloupka, Disertační práce, *Rozpoznávání akustického signálu řeči s podporou vizuální informace*, 2005

[6] Cambridge University Engineering Department, The HTK Book, First published December 1995, Revised for HTK Version 3.2 December 2002disp

[7] Vladimír Myslík, Neuronové sítě, [Online; navštíveno 28.4. 2008]. Dostupné z: `http://aldebaran.feld.cvut.cz/~xmyslik/www/neural.html`

[8] Sonka,M., Hlaváč V., Boyle R.: Image Processing, Analysis, and Machina Vision, *In PWS Publishing*, 1998, ISBN 0-534-95393-X

[9] Technická univerzita Ostrava, Umělé neuronové sítě, [Online; navštíveno 29.4. 2008]. Dostupné z: `http://www.fs.vsb.cz/books/NeuronoveSite/NN_pojmy.htm`

[10] České vysoké učení technické, Neuronové sítě, [Online; navštíveno 29.4. 2008]. Dostupné z: `http://cyber.felk.cvut.cz/gerstner/biolab/bio_web/teach/FunBio/neuron/neursite.html`

[11] Masarykova univerzita, Skryté Markovovy modely, [Online; navštíveno 2.5. 2008]. Dostupné z: http://nlp.fi.muni.cz/nlp/nlp-prace/referaty/xkrivan/HMM.html

[12] Intel Corporation, Microcomputer Research Labs, A coupled HMM for audio-vizual sérech recognition, [Online; navštíveno 29.4. 2008]. Dostupné z: http://www.cs.ubc.ca/~murphyk/Papers/icassp02.pdf

[13] Západočeská univerzita v Plzni, Audiovizuální rozpoznávání řeči [Online; navštíveno 15.3. 2008]. http://www.kky.zcu.cz/cs/research-fields/audio-visual-speech-recognition

Příloha č. 1 – Rozpoznávací skóre při použití DCT příznaků

1. sloupec–druh příznaků, 2. sloupec–počet stavů, 3. sloupec–úspěšnost rozpoznání [%]

stat	1	6,0
stat	2	12,4
stat	3	13,2
stat	4	18,4
stat	5	22,0
stat	6	20,8
stat	7	20,4
stat	8	25,6
stat	9	26,0
stat	10	26,4
stat	11	30,0
stat	12	30,8
stat	13	29,6
stat	14	32,4
dyn	1	12,4
dyn	2	18,4
dyn	3	20,8
dyn	4	25,6
dyn	5	26,4
dyn	6	30,8
dyn	7	32,4
dyn	8	6,4
dyn	9	14,8
dyn	10	23,6
dyn	11	26,7
dyn	12	30,0
dyn	13	29,2
dyn	14	32,0
akc	1	13,2
akc	2	20,8
akc	3	26,0
akc	4	30,8
akc	5	6,0
akc	6	14,8
akc	7	19,6
akc	8	30,0
akc	9	31,2
akc	10	2,8
akc	11	8,8
akc	12	8,8
akc	13	10,0
akc	14	10,0
stat+dyn	1	18,4
stat+dyn	2	25,6
stat+dyn	3	30,8
stat+dyn	4	6,4
stat+dyn	5	23,6
stat+dyn	6	30,0
stat+dyn	7	32,0
stat+dyn	8	6,8
stat+dyn	9	8,8
stat+dyn	10	10,0
stat+dyn	11	12,8
stat+dyn	12	24,8
stat+dyn	13	36,0
stat+dyn	14	40,0
stat+akc	1	22,0
stat+akc	2	26,4
stat+akc	3	6,0
stat+akc	4	23,6
stat+akc	5	30,4
stat+akc	6	2,8
stat+akc	7	7,6
stat+akc	8	10,4
stat+akc	9	16,0
stat+akc	10	30,8
stat+akc	11	38,8
stat+akc	12	14,4
stat+akc	13	28,8
stat+akc	14	38,0
dyn+akc	1	20,8
dyn+akc	2	30,8
dyn+akc	3	14,8
dyn+akc	4	30,0
dyn+akc	5	2,8
dyn+akc	6	8,8
dyn+akc	7	10,0
dyn+akc	8	24,8
dyn+akc	9	36,8
dyn+akc	10	14,4
dyn+akc	11	30,8
dyn+akc	12	5,2
dyn+akc	13	21,2
dyn+akc	14	28,4
stat+dyn+akc	1	20,4
stat+dyn+akc	2	32,4
stat+dyn+akc	3	19,6
stat+dyn+akc	4	32,0
stat+dyn+akc	5	7,6
stat+dyn+akc	6	10,0
stat+dyn+akc	7	25,6
stat+dyn+akc	8	40,4
stat+dyn+akc	9	25,6
stat+dyn+akc	10	38,0
stat+dyn+akc	11	16,8
stat+dyn+akc	12	28,4
stat+dyn+akc	13	30,4
stat+dyn+akc	14	43,6

Příloha č. 2 –Rozpoznávací skóre při použití geometrických příznaků

1. sloupec–druh příznaků, 2. sloupec–počet stavů, 3. sloupec–úspěšnost rozpoznání [%]

stat	1	8,8	stat+dyn	8	2,4
stat	2	11,2	stat+dyn	9	4,8
stat	3	14,8	stat+dyn	10	6,4
stat	4	17,6	stat+dyn	11	15,2
stat	5	18,0	stat+dyn	12	24,4
stat	6	18,0	stat+dyn	13	29,2
stat	7	19,2	stat+dyn	14	33,6
stat	8	21,6	stat+akc	1	18,0
stat	9	19,6	stat+akc	2	22,4
stat	10	22,4	stat+akc	3	5,2
stat	11	23,2	stat+akc	4	12,0
stat	12	24,0	stat+akc	5	16,0
stat	13	27,6	stat+akc	6	3,6
stat	14	26,4	stat+akc	7	5,6
dyn	1	11,2	stat+akc	8	6,4
dyn	2	17,6	stat+akc	9	17,6
dyn	3	18,0	stat+akc	10	29,2
dyn	4	21,6	stat+akc	11	33,6
dyn	5	22,4	stat+akc	12	18,8
dyn	6	24,0	stat+akc	13	23,2
dyn	7	26,4	stat+akc	14	29,2
dyn	8	6,0	dyn+akc	1	18,0
dyn	9	8,0	dyn+akc	2	24,0
dyn	10	12,0	dyn+akc	3	8,0
dyn	11	12,4	dyn+akc	4	17,6
dyn	12	17,6	dyn+akc	5	3,6
dyn	13	24,0	dyn+akc	6	4,8
dyn	14	20,4	dyn+akc	7	8,4
akc	1	14,8	dyn+akc	8	24,4
akc	2	18,0	dyn+akc	9	30,4
akc	3	19,6	dyn+akc	10	18,8
akc	4	24,0	dyn+akc	11	26,8
akc	5	5,2	dyn+akc	12	3,6
akc	6	8,0	dyn+akc	13	10,8
akc	7	10,8	dyn+akc	14	21,2
akc	8	17,6	stat+dyn+akc	1	19,2
akc	9	17,6	stat+dyn+akc	2	26,4
akc	10	3,6	stat+dyn+akc	3	10,8
akc	11	4,0	stat+dyn+akc	4	20,4
akc	12	4,8	stat+dyn+akc	5	5,6
akc	13	6,8	stat+dyn+akc	6	8,4
akc	14	8,4	stat+dyn+akc	7	26,4
stat+dyn	1	17,6	stat+dyn+akc	8	31,2
stat+dyn	2	21,6	stat+dyn+akc	9	19,2
stat+dyn	3	24,0	stat+dyn+akc	10	29,2
stat+dyn	4	6,0	stat+dyn+akc	11	10,0
stat+dyn	5	12,0	stat+dyn+akc	12	21,2
stat+dyn	6	17,6	stat+dyn+akc	13	25,6
stat+dyn	7	20,4	stat+dyn+akc	14	32,4