VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Magistrinis darbas

Kalbos atpažinimas kompiuteriu (Speech recognition by computer)

Atliko: 2 kurso 1 grupės studentas
Justinas Bardauskas .....
(parašas)

Darbo vadovas:
doc. dr. Algirdas Bastys.....
(parašas)

Vilnius
2012

Turinys

Summary

This work focuses on speech recognition by computer, pattern recognition stages and problems. Also there is a goal to create a speech recognition tool.

At the beginning, there is a general overview of the audio signal and language concepts. The subsequent presentation of the essential tasks of speech recognition, introduction to matrix algebra, which is used to described algorithm.

Information is provided on what basis and how features are extracted. For this work often is used LPC. This algorithm is one of the most popular extracting features of speech signal, so it is reviewed in this paper, as well as its modification WLPC.

The following text of the speech recognition gives theory of extracted features use. Section „Acoustic modeling“ describes the recognition of speech units and one of the most commonly used acoustic modeling technologies – Hidden Markov Models and the next section „Speech modeling“ describes the language modeling, which purpose is to correct data referring to dictionaries and speech structure.

The rest of the text is focused on speach recognition using specrtogram and implementation of speach recognition system. After that, there were executed experiments, that where used to define quality of speech recognition.

Anotacija

Šiame darbe gilinamasi į kalbos atpažinimą kompiuteriu, atpažinimo etapus, problemas, o vėliau mėginama sukurti kalbos atpažinimo įrankį.

Pradžioje, bendrai apžvelgiama garso signalo, kalbos sąvokos. Vėliau pateikiamos kalbos atpažinimo esminiai uždaviniai, supažindinama su matricų algebra, kuri naudojama aprašytuose algoritmuose.

Pateikiama informacija kuo remiantis ir kaip išskiriami požymiai. Šiam darbui dažnai naudojamas LPC. Šis algoritmas yra vienas iš populiariausių išskiriant kalbos signalo požymius, todėl jis šiame darbe yra apžvelgtas, kaip ir jo modifikacija WLPC.

Toliau tekste pateikiama kalbos atpažinimo teorija, apie išskirtų požymių panaudojimą. Skyriuje „Akustinis modeliavimas“, aprašomas kalbos vienetų atpažinimas ir vienas iš dažniausiai naudojamų akustinio modeliavimo technologijų - paslėptieji Markov'o modeliai, sekantis skyrius „Kalbos modeliavimas“, aprašo kalbos modeliavimą, skirtą jau turimiems duomenims sutvarkyti, remiantis žodynais ir analizuojamos kalbos struktūra.

Likusioje teksto dalyje koncentruojamasi ties kalbos atpažinimu panaudojant spektrogramą ir kalbos atpažinimo sistemos įgyvendinimu. Po to atlikti eksperimentai, kuriais buvo tiriama pateikto algoritmo atpažinimo kokybė.

Įvadas

Šiame darbe bus apžvelgiama tema „Kalbos atpažinimas kompiuteriu“, iš ko susideda, kokios yra pagrindinės kalbos atpažinimo problemos, kokie egzistuoja šiuolaikiniai šių problemų sprendimo būdai, išsiaiškinti kokie yra pagrindiniai kalbos atpažinimo etapai. Susipažinti su *LPC(Linear Prediction Coefficients)* ir *WLPC(Warped LPC)* požymių išskyrimo algoritmais ir *HMM(Hidden Markov Model)* modeliu, jo panaudojimu kalbos atpažinime. Įsigilinama į kalbos atpažinimą naudojant kalbos spektrogramą, kuri panaudota praktinėje dalyje. Taip pat pateikiama informacija apie sukurtą kalbos atpažinimo dizainą ir atliktus darbus įgyvendinant kompiuterinę programą.

Pradžioje supažindinimą su tiriamuoju objektu, aprašoma kalbos savoka ir keliamos problemos, su kokiomis susiduriama, norint atpažinti žmogaus kalbą.

Kadangi kalbos atpažinime naudojami įvairūs algoritmai, kurie yra aprašyti matematiškai ir norint paaiškinti juos, reikia susipažinti su tam tikromis matematinėmis operacijomis, todėl yra parašytas skyrelis, kur pateikiama informacija apie veiksmus su matricomis, kurie naudojami tiesinės prognozės metoduose. Po to aprašoma ir tiesinė prognozė ir ja paremti algoritmai tokie kaip LPC ir WLPC. Taip pat pateikiama informacija apie kalbos signalo analizę, kurioje apžvelgiama signalo spektrogramos ir jų teikiami duomenys.

Vėliau aprašomas bendras kalbos atpažinimo procesas. Šioje dalyje kalbama apie akustinį modeliavimą, kurio paskirtis sukurti kalbos atpažinimo modelį, pagal kurį bus identifikuojami kalbos vienetai(fonemos, žodžiai, frazės), apie paslėptąjį Markov'o modelį (HMM), kurio paskirtis kalbos atpažinime yra akustinio modelio sukūrimas ir apie kalbos(struktūros) modeliavimą, kurio tikslas remiantis žodynu ir kalbos struktūra surasti neatpažintus žodžius ir juos pakeisti labiausiai tikėtinais ir pan., ir pabaigoje pateikiama viena iš galimų kalbos atpažinimo schemų bei jos žingsnių aprašymas.

Ir galiausiai pateikiama praktinė dalis, kurioje pateikiama planuoti darbai, sukurtas programos dizainas ir praktiškai atlikti darbai.

Šio darbo motyvas yra įsigilinti į kalbos atpažinimą naudojantis kompiuteriu ir praktiškai įgyvendinti kalbos atpažinimo programą, gebančią atpažinti izoliuotas frazes.

1. Garsas

Kalba sudaryta ir garsinių signalų, kuriuos suprasdami galime bendrauti. Paprastai tariant, žmogus kalba frazėmis, o frazės sudaromos iš žodžių garsų, o šie savo ruoštu sudaromi iš raidžių garsų. Toks garsas vadinamas **fonema**. Pats garsas yra slenkantis slėgio svyravimas terpėje, kuris yra girdimas žmogaus ar gyvūno ausimis [wiki11b]. Žmogaus ausimi girdimų garsų dažnis yra intervale 16 Hz – 20 000 Hz.

1.1. Garso savybės

Garsas apibūdinamas trimis dydžiais: *stipriu, aukščiu ir tembru* [CF00].

Pagal *garso stiprį* atskiriame stiprų garsą nuo silpno. Garso stipris - tai dydis, nusakantis energijos kiekį, kurį garso banga per vienetinį laiką perneša pro vienetinį paviršių, statmeną bangos sklidimo krypčiai.

Garso aukštis - tai garso savybė, leidžianti atskirti duslius garsus nuo aštrių. Didėjant garso bangos dažniui, garso aukštis didėja.

Garso tembras - instrumentų, balsų ar bet kokių kitų garso šaltinių skambesio savitumas. Tembras nusako muzikinio garso skambesio pobūdį. Jis padeda klausytojui atskirti skirtingos kilmės garsus, pvz.: balsą nuo muzikos instrumentų. Fizikiniai garso duomenys, keičiantys tembro supratimą, yra spektras ir dažnių kreivė.

1.2. Kaip atsiranda garsas ?

Garsas yra tam tikra kinetinės energijos (judesio energijos) forma, kurią sukuria bet kuris virpantis objektas. Visų garsų priežastis yra mechaniniai aplinkos virpesiai, nors paprastai jie nematomi. Pavyzdžiui, kalbame ir dainuojame virpant balso stygoms gerklose [wiki11b].

Virpantis kūnas verčia virpėti arčiausiai esančias oro molekules. Virpesiai sklinda ore, sudarydami garso bangą, tačiau oras banga neslenka. Ten kur molekulės susispiečia, susidaro didesnio slėgio sritis (sutankėjimas), o ten, kur jų lieka mažiau — žemesnio slėgio sritis (praretėjimas). Pakaitomis einančios sutankėjimų ir praretėjimų sritys sklinda ore kaip garso banga. Pasiekus ausį, ji virpina ausies būgnelį, ir žmogus girdi garsą [wiki11b].

Kuo stipresni daikto virpesiai, tuo didesnis sutankėjimų ir praretėjimų slėgio skirtumas ir garsas tuo stipresnis. Virpesių dažnis lemia garso aukštį, arba toną. Jei virpesiai dažnėja, sutankėjimų ir praretėjimų sritys suartėja — garsas darosi aukštesnis. Lėtesni virpesiai atitinka žemesnį garsą [wiki11b].

1.3. Triukšmas ir akustika

Triukšmas – įvairaus pobūdžio nepageidaujamas garsas (akustinis triukšmas). Kalbant apie triukšmą kaip garsą, paprastai turima omenyje beprasmis (girdinčiojo atžvilgiu) ir stipresnis nei įprasta garsas. Kokio pobūdžio garsas yra interpretuojamas kaip triukšmas, priklauso ir nuo interpretuojančiojo santykio su tuo garsu. Aukšto dažnio garsas suvokiamas

kaip labiau triukšmingas nei žemo dažnio (nors žemo dažnio triukšmas yra kenksmingesnis žmogaus sveikatai). Triukšmas, kurio stiprumas kinta, taip pat suvokiamas kaip stipresnis nei pastovaus intensyvumo garsas. Triukšmas atrodo silpnesnis, kai girdintysis gali lokalizuoti triukšmo šaltinį. Taip pat didesniu triukšmu laikomi garsai, kurie tą patį stiprumą pasiekia per trumpesnį laiką (staigus garsumo padidėjimas) [wiki11c].

Garso, vaizdo bei transliacijos sistemose garsiniu (audio) triukšmu vadinamas foninis garsas (šnypštimas, ūžimas, zvimbimas), kuris išryškėja tyliose programos dalyse. Pagal fizines charakteristikas toks triukšmas skirstomas pavadinant atitinkamomis spalvomis, remiantis apytikre analogija tarp garso dažnių ir spalvų bangų dažnių spektrų (pvz., baltasis triukšmas, rausvasis triukšmas) [wiki11c].

Garso inžinerijoje triukšmas taip pat reiškia elektroninį triukšmą, sukeliantį šnypštimo efektą [wiki11c].

2. Kalba

Kalba – garsinis signalas, kuris sukuriamas kalbos aparato (pvz.: burna, liežuvis, garso stygos). Signalas gali būti diskretus (pvz.: simboliai iš abėcėlės) arba nepertraukiamas, tolydūs (pvz.: kalbos pavyzdžiai, temperatūros matavimai, muzika, ...). Signalų šaltinis gali būti pastovus arba nepastovus. Žmonėms šie signalų tipai nėra labai aktualūs, nes sugeba gerai suprasti juos, netgi su dideliu triukšmu ar net iškraipytą įrašą daugeliu atvejų žmogui atpažinti kas sakoma nėra sunku [Rab89].

Kiekvieną posakį galime įsivaizduoti kaip tiesę, kurią skirstome segmentais. Mažiausias funkciškai savarankiškas tiesinis kalbos segmentas vadinamas *fonema*. Fonema yra abstraktus vienetas, kuris kalboje realizuojamas kaip konkretus garsas. Funkcinį garsų savarankiškumą rodo tai, kad jie padeda skirti žodžius. Pavyzdžiui, žodžiai *tarė* ir *darė* fonetiškai skiriasi pirmaisiais priebalsiais. Vadinasi, *t* ir *d* funkcionuoja kaip skirtingos fonemos. Jos atlieka distinktyvinę (skiriamąją) funkciją [Urb11].

Tačiau kad žmogus galėtų šiuos garsus ištarti, jam reikia juos *artikuliuoti* t.y. naudojantis kalbos aparatu (burna, liežuviu, garso stygomis) generuoti raidžių garsus (fonemas). Šis tarimas sukelia kitą efektą vadinamą *koartikuliacija* arba *antrine artikuliacija*. *Koartikuliacija* – fonemų (garsų) akustinė realizacija priklauso nuo gretimų fonemų [Bas04], t. y. tariant žodį, tame žodyje esantys garsai persidengia, ar kitaip užgožia, pakeičia kitą garsą. Dėl šio efekto, kalbos atpažinimas kompiuteriu tampa tik sudėtingesnis, nes tai apsunkina fonemų išskyrimą kalboje.

3. Kalbos atpažinimo problemos

Daugeliu atvejų žmogui suprasti, ką kalba kitas žmogus triukšmingoje aplinkoje, ar klausantis muzikos, ar prastos kokybės įrašo, nėra sunku. Tačiau kompiuteriui tai milžiniškas darbas. Kalba - tai analoginis garsinis signalas. Kompiuteris dirba tik su skaimenizuotais duomenimis.

Pirma problema, reikia šios signalus skaitmenizuoti. Šiais laikais tai jau ne be problema, nes egzistuoja garso kortos, kurios su šiuo darbu kuo puikiausiai susitvarko.

Antra problema yra gautų duomenų analizė. Kalba yra sudėtinga garsinių signalų aibė, kuri turi savo struktūrą, savus požymius, pagal kuriuos galima mėginti analizuoti. Pati analizė, susideda iš kelių esminių komponentų: duomenų gavimo, duomenų paruošimo analizei, duomenų požymių radimo. Kadangi kalba – sudėtingas įrankis bendravime, tai norint jį tinkamai panaudoti kompiuterijos srityje, reikia gerai išsiaiškinti, iš ko susideda kalba, pagal kokius požymius galima atpažinti vieną ar kitą žodį ar frazę ir pan.

Trečia problema labiau susijusi jau su pačiu kalbos atpažinimo procesu. Tariami žodžiai ar frazės gali būti atskiros (diskrečios, t.y. sakomos su pauzėmis) ir gali būti vientisos (pvz.: normali žmogaus kalba). Pirmu atveju išskirti žodį ar frazę nėra sudėtinga, nes duomenys yra pavieniai žodžiai ar frazės, kur žodžio pradžia ir pabaiga atskiriama pauzėmis. Antru atveju situacija yra kur kas sudėtingesnė. Žodžiai ar frazės nuo kitų žodžių ar frazių nėra atskirti pauzėmis ar kitais garsais, anaiptol, vientisoje kalboje žodžiai yra susipynę tarpusavyje dėl koartikuliacijos efekto. Norint išspręsti šia problemą yra kuriami įvairūs akustiniai ir kalbos modeliai, kuriais remiantis mėginama išskirti žodžius ar frazes.

Ketvirta problema – kaip atlikti efektyviai iškirpto žodžio, frazės paiešką žodyne. Pats žodyno sudarymas, nėra paprastas darbas. Norint, kad kalbos atpažinimo programa veiktų su realia šnekamąja kalba reikia, kad žodyną sudarytų 10 - 20 tūkstančių įrašų. Žodyno struktūra ir esamų reikšmių pateikimas priklauso nuo pačios sistemos įgyvendinimo, taip kad bendro standarto nėra.

Šios išvardintos problemos ir sudaro kalbos atpažinimą. Taigi, šių problemų sprendinys ir yra kalbos atpažinimo sistema.

4. Matricų algebra tiesinei prognozei

Kadangi šiame tekste naudojamos matematinės formulės bei veiksmai, reikia apžvelgti tai, kad tolimesnis tekstas būtų visiškai suprantamas. Šiame skyrelyje bus pateikta matrcų algebrą, kuri aprašo Tioplico ir sąsukos matricas, Levinson–Durbin rekursiją. Šioje dalyje aprašyti metodai yra paimti iš [Bac04].

4.1. Tioplico matricos

Tioplico (ang. Teoplitz) matricos - tai matricų klasė, kuri dažnai atsiranda skaitmeninių signalų apdorojime. Charakteristinė šių matricų savybė yra tokia, kad matricos gretimos eilutės ir stulpeliai yra panašūs ir skiriasi tik poslinkiu. Kitais žodžiais tariant, $n \times k$ Tioplico matrica $\mathbf{T}$ apibrėžiama kaip $\mathbf{T}_{ij} = t_{j-i}$ arba matricos forma

$$\mathbf{T} = \begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_{k-1} \\ t_{-1} & t_0 & t_1 & \ddots & t_{k-2} \\ t_{-2} & t_{-1} & t_0 & \ddots & t_{k-3} \\ \vdots\ddots & \ddots & \ddots & \ddots & \vdots \\ t_{-n+1} & t_{-n+2} & t_{-n+3} & \ddots & t_{k-n-2} \end{bmatrix}.$$

Tioplico matricos taip pat priklauso per-simetriniai(ang. persymmetric) matricų klasei, kur matricos įstrižainė yra simetrinė savo šiaurės rytų - pietvakarių kryptimis. Tioplico matricos daugeliu atvejų yra ekvivalenčios žiedinės (ang. circulant) matricos. Žiedinės matricos kiekviena eilutė gaunama ciklišk ai pastumiant buvusią eilutę į dešinę.

Skaitmeninių signalų apdorojime dažniausiai pasitaiko simetrinės Tioplico matrico. Simetrinė matrica $\mathbf{A}$ turi savybę $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ ir dėl to simetrinė Tioplico matrica turės $\mathbf{T}_{ij} = t_{|j-i|}$. Be to, dar reikia apriboti matricas apibrėžiant kaip realias ir teigiamas, nes jos, praktinių požiūriu, vienos iš dažniausiai naudojamų kalbos atpažinime. Teigiamai apibrėžtos matricos $\mathbf{A}$, pagal apibrėžimą, yra tokios $\mathbf{x}^T$ kurios $\mathbf{Ax} \geq 0$ visiems $\mathbf{x} \in \mathbb{R}^n$ ir $\mathbf{x}^T\mathbf{Ax} = 0$, tai reiškia $\| \mathbf{x} \| = 0$.

4.2. Levinson-Durbin rekursija

Daugelyje aplikacijų, kur Tioplico matricos naudojamos, susiduriama su lygtimi

$$\mathbf{Ta} = \mathbf{b}, \tag{1}$$

kur $\mathbf{T}$ yra Tioplico matrica $n \times n$, o $\mathbf{a}$ ir $\mathbf{b}$ yra vektoriai. Uždavinys yra toks: reikia surasti $\mathbf{a}$, kai $\mathbf{T}$ ir $\mathbf{b}$ yra žinomi.

Pradžiai, apibrėžiama pagrindinė sub-matrica $\mathbf{T}_p$, kaip matricos $\mathbf{T}$ viršutinis kairysis

blokas. Toliau teigiama, kad turimas sprendinys p eilės $\mathbf{a}_p$, kurio lygtis

$$\begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_p \\ t_1 & t_0 & t_1 & \ddots & t_{p-1} \\ t_2 & t_1 & t_0 & \ddots & t_{p-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_p & t_{p-1} & t_{p-1} & \ddots & t_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2}$$

Papildoma matrica $\mathbf{a}_p$ nuliais

$$\begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_p \\ t_1 & t_0 & t_1 & \ddots & t_{p-1} \\ t_2 & t_1 & t_0 & \ddots & t_{p-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_p & t_{p-1} & t_{p-1} & \ddots & t_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \\ 0 \end{bmatrix} = \begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \eta_p \end{bmatrix}, \tag{3}$$

kur $\eta_p = \sum_{i=0}^{p} a_i^{(p)} t_{p-i+1}$ ir $a_0^{(p)} = 1$. Pagrindinis žingsnis šioje realizacijoje yra simetrinis, kai $\mathbf{T}_p\mathbf{a}_p = \mathbf{u}_p$, kur $\mathbf{u}_p = [\epsilon_p \dots 0]^T$ ir $\mathbf{T}_p$. Gautas rezultatas yra $\mathbf{T}_p\mathbf{a}_p^{\#} = \mathbf{u}_p^{\#}$, kur simbolis $\#$ žymi eilutės apvertimą. Pažymint atspindžio koeficientą Γ_p, gaunama

$$\mathbf{T}_{p+1}\left(\begin{bmatrix} 1 \\ a_1^{(p)} \\ \vdots \\ a_p^{(p)} \\ 0 \end{bmatrix} + \Gamma_p \begin{bmatrix} 0 \\ a_p^{(p)} \\ a_{p-1}^{(p)} \\ \vdots \\ 1 \end{bmatrix}\right) = \left(\begin{bmatrix} \epsilon_p \\ 0 \\ \vdots \\ 0 \\ \eta_p \end{bmatrix} + \Gamma_p \begin{bmatrix} \eta_p \\ 0 \\ \vdots \\ 0 \\ \epsilon_p \end{bmatrix}\right). \tag{4}$$

Γ_p pasirenkamas toks, kad $\eta_p + \Gamma_p\epsilon_p = 0$ duotų $p+1$ eilės sprendinį lygčiai (2) kaip

$$\mathbf{a}_{p+1} = \begin{bmatrix} \mathbf{a}_p \\ 0 \end{bmatrix} + \Gamma_p \begin{bmatrix} 0 \\ \mathbf{a}_p^{\#} \end{bmatrix}. \tag{5}$$

Taigi su tinkamomis pradinėmis reikšmėmis ($\mathbf{a}_0 = 1$) ši procedūra gali būti naudojama rekursyviai spręsti uždavinį (2). Be to, naudojant tarpines reikšmes $\mathbf{a}_p$ galima spręsti lygtis, kurių tipas yra (1). Ankščiau aprašytas algoritmas, skirtas spręsti uždaviniui (2), vadinamas *Levinson-Durbin rekursija*, o vėlesnis skirtas spręsti pasirenkamas lygtis, kurių tipas yra (1), vadinamas *Levinson rekursija*. Šio algoritmo sudėtingumas yra $O(n^2)$.

4.3. Sąsukų matrica

Sąsukų (ang. convolution) matricos $\mathbf{A}$ polinomas $A(z) = \sum_{i=0}^{m-1} a_i z^{-i}$ apibrėžiamas kaip matrica

$$\mathbf{A}^T = \begin{bmatrix} a_0 & a_1 & \dots & a_{m-1} & 0 & 0 & \dots & 0 \\ 0 & a_0 & a_1 & \dots & a_{m-1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_0 & a_1 & \dots & a_{m-1} & 0 \\ 0 & 0 & \dots & 0 & a_0 & a_1 & \dots & a_{m-1} \end{bmatrix}.$$

Pavadinimas, sąsukų matrica, atsiranda iš fakto, kad vektorių daugyba $\mathbf{b} = [b_0 b_1 ... b_n]^T$ su $\mathbf{A}$, pvz. $\mathbf{A}^T\mathbf{b}$ atitinka sąsukų sekas b_n ir a_n. Tai yra ekvivalentu polinomų daugybai $B(z) = \sum_{i=0}^{m} b_i z^{-i}$ su $A(z)$, pvz. $A(z)B(z)$.

5. Tiesinė prognozė

Šis skyrius parašytas pagal autoriaus [Bac04] darbą.

Tarkime, kad turime signalo x_n praėjusių signalo mėginių (ang. samples) m intervale $[n-m, n-1]$ ir užduotis yra apytikriai apskaičiuoti mėginį x_n. Apytikris įvertis $\hat{x}_n$ gali būti apibrėžtas kaip

$$\hat{x}_n = -\sum_{i=1}^{m} a_i x_{n-i}, \tag{6}$$

kur minuso ženklas buvo pridėtas dėl patogumo ir a_i $(1 \leq i \leq m)$ yra modelio parametrai. Paklaida e_n tarp apskaičiuoto mėginio ir tikro mėginio x_n yra tokia:

$$e_n = x_n - \hat{x}_n = x_n + \sum_{i=1}^{m} a_i x_{n-1} = \sum_{i=0}^{m} a_i x_{n-1} = \mathbf{a}^T \mathbf{x}, \tag{7}$$

su apribojimais $a_0 = 1$, $\mathbf{a}^T = [a_0 ... a_m]$ ir $\mathbf{x}^T = [x_0 ... x_{n-m}]$. Tikslas yra surasti geriausią įmanomą įvertį, kad parametrų aibė a_i, tokių, kurios paklaida e_n yra mažiausia. Dažnas optimizavimo kriterijus yra *bendra kvadratinė paklaida*

$$\alpha = \sum_{n=n_0}^{n_1} = \sum_{n=n_0}^{n_1} \sum_{i=0}^{m} \sum_{j=0}^{m} a_i x_{n-i} xn - j a_j = \mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{a}^T \mathbf{C}_\mathbf{x} \mathbf{a}, \tag{8}$$

kur $\mathbf{X}$ yra $(n_1 - n_0 + 1) \times m$ sąsukų matrica sudaryta iš x_n ir kovariacijų matrica $\mathbf{C}_x$ apibrėžiama kaip

$$\mathbf{C}_x = \mathbf{X}^T \mathbf{X}. \tag{9}$$

Tam, kad minimizuoti α, reikia apriboti $a_0 = 1$, kuris gali gaunamas naudojant Lagranžo (ang. Lagrange) daugiklį λ su funkcija

$$\eta(\mathbf{a}, \lambda) = \mathbf{a}^T \mathbf{C}_x \mathbf{a} - 2\lambda \mathbf{a}^T \mathbf{u}, \tag{10}$$

kur $\mathbf{u} = [100...0]^T$.

Minimumas randamas prilyginant dalinę išvestinę $\partial / \partial_{a_k}$ nuliui ir tuomet gaunama lygtis

$$0 = \frac{\partial \eta(\mathbf{a}, \lambda)}{\partial \mathbf{a}} = 2\mathbf{C}_x \mathbf{a} - 2\lambda \mathbf{u}, \tag{11}$$

kurios sprendinys yra

$$\mathbf{C}_x \mathbf{a} = \lambda \mathbf{u}. \tag{12}$$

Prognozės klaida $\mathbf{e} = [e_{n_0} e_{n_0+1} ... e_{n_1}]$ randama pagal

$$\mathbf{e} = \mathbf{X}\mathbf{a}. \tag{13}$$

Šis metodas bendrai žinomas kaip *kovariacijų (ang. covariance) metodu.*

Kitas matematiškai patvirtintas minimizavimo kriterijus yra minimizuoti reikšme $E[\cdot]$,

sudarytą iš kvadratinės paklaidos e_n^2. Remiantis ankstesnėmis formulėmis gaunama

$$E[e_n^2] = E[\mathbf{a^T x^T} xa] = \mathbf{a^T} E[\mathbf{x^T} x]a = \mathbf{a^T R a}, \tag{14}$$

kur $\mathbf{R}$ yra autokoreliacijos matrica, kuri yra reali, simetrinė ir Tioplico (ang. Toeplitz). Kaip kovoriacijos metode, turim funkcinį sprendimą

$$\eta(\mathbf{a}, \lambda) = \mathbf{a}^T \mathbf{R a} - 2\lambda \mathbf{a}^T \mathbf{u}, \tag{15}$$

kuris pateikia lygtis

$$\mathbf{Ra} = \sigma^2 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{16}$$

kur $\sigma^2 = \lambda$ paklaidos energija, nes

$$E[e_n^2] = \mathbf{a}^T \mathbf{R a} = \sigma^2 \mathbf{a}^T \mathbf{u} = \sigma^2 a_0 = \sigma^2.$$

Šis metodas žinomas kaip *autokoreliacijos (ang. autocorrelation) metodas.*

5.1. Iškraipyta tiesinė prognozė

Iškraipymas - tai technika, kuri padidina modelio dažnių skiriamąją gebą transformuojant į „iškraipytą“ dažnių skalę, kur modelis yra konstruojamas. Atvirkštinė (ang. inverse) transformacija atitinka originalių dažnių sritį. Motyvacija yra tokia, jeigu iškreipta dažnių sritis turi dažnių skiriamąją gebą, kuri yra panaši į tą, kuri nuspėjama, tuomet sugeneruotas modelis iškraipytoje srityje paims tas signalo charakteristikas, kurios suvokiamos kaip svarbios.

Iškraipytos tiesinės prognozės (ang. warped linear prediction) modelis gaunamas pirmiausiai pritaikant Z-transformaciją išraiškai iš skyrelio „Tiesinis prognozavimas“

$$\hat{x}_n = -\sum_{i=1}^{m} a_i x_{n-1}, \tag{17}$$

iš kurios gaunama

$$\hat{X} = -\sum_{i=1}^{m} a_i z^{-i} X(z). \tag{18}$$

Tuomet pakeičiamas uždelsimas z^{-i} tiesiniu „all-pass“ filtru $D(z)$ ir gaunam

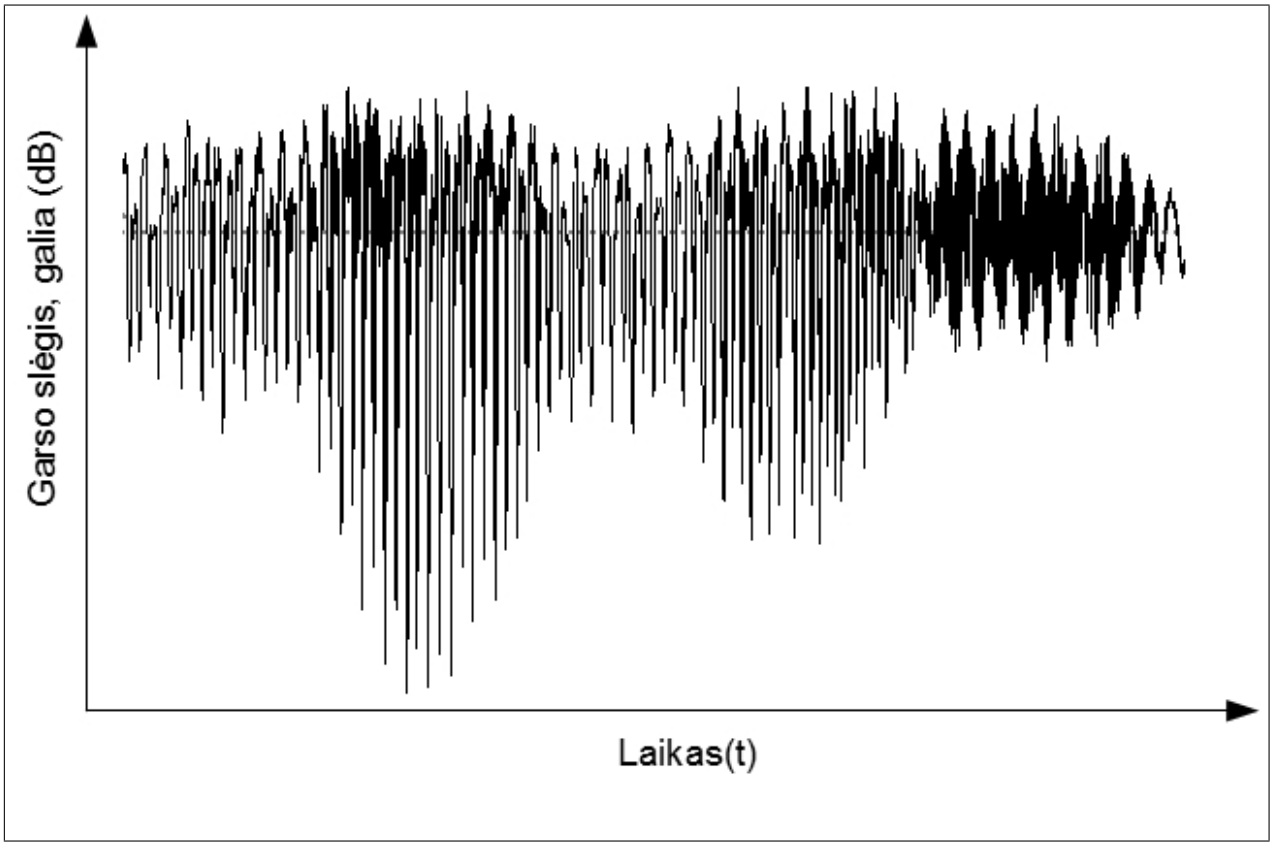
$$\hat{X} = -\sum_{i=1}^{m} a_i D^i(z) X(z). \tag{19}$$

Dažnas iškraipymo metodas naudoja visus dažnius praleidžiantį (ang. all-pass) modelį

$$D(z) = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}}. \tag{20}$$

6. Kalbos signalo analizė

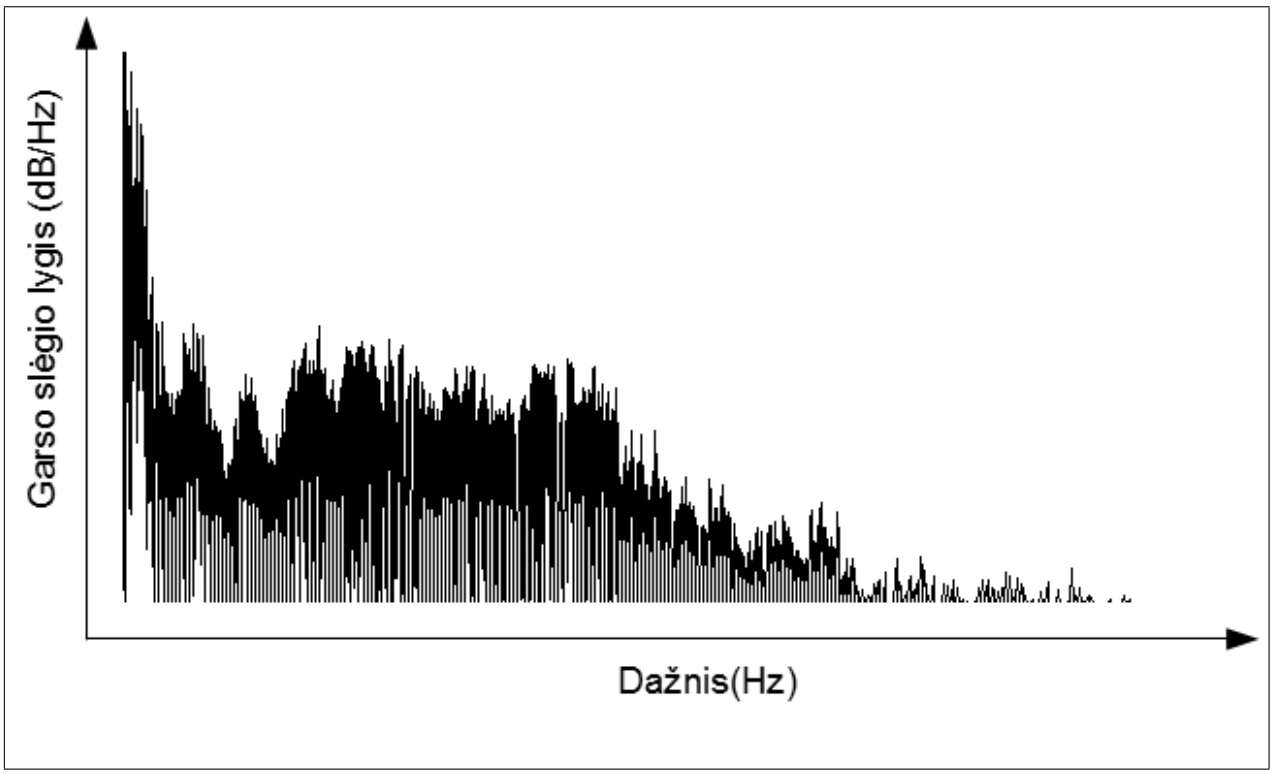
Kalba yra signalas, tačiau kad jį galima būtų analizuoti, reikia kažkokiu suprantamu būdu jį pateikti. Tam tikslai naudojama signalo bangos forma, signalo spektras ir spektrograma. Atvaizduojant garso signalą kaip bangos forma (1 pav. 15 psl.) naudojama laiko ir garso galios arba slėgio skalėje. Tokia reprezentacija pateikiama kaip garso slėgio ar galios pokyčio laike grafikas. Garso galia ar slėgis matuojamas decibelais, o dažnis matuojamas vibracijomis per sekunde (hercais).



1 pav.: Ištarto žodžio „vienas“ garso signalo bangos forma (ang. waveform).

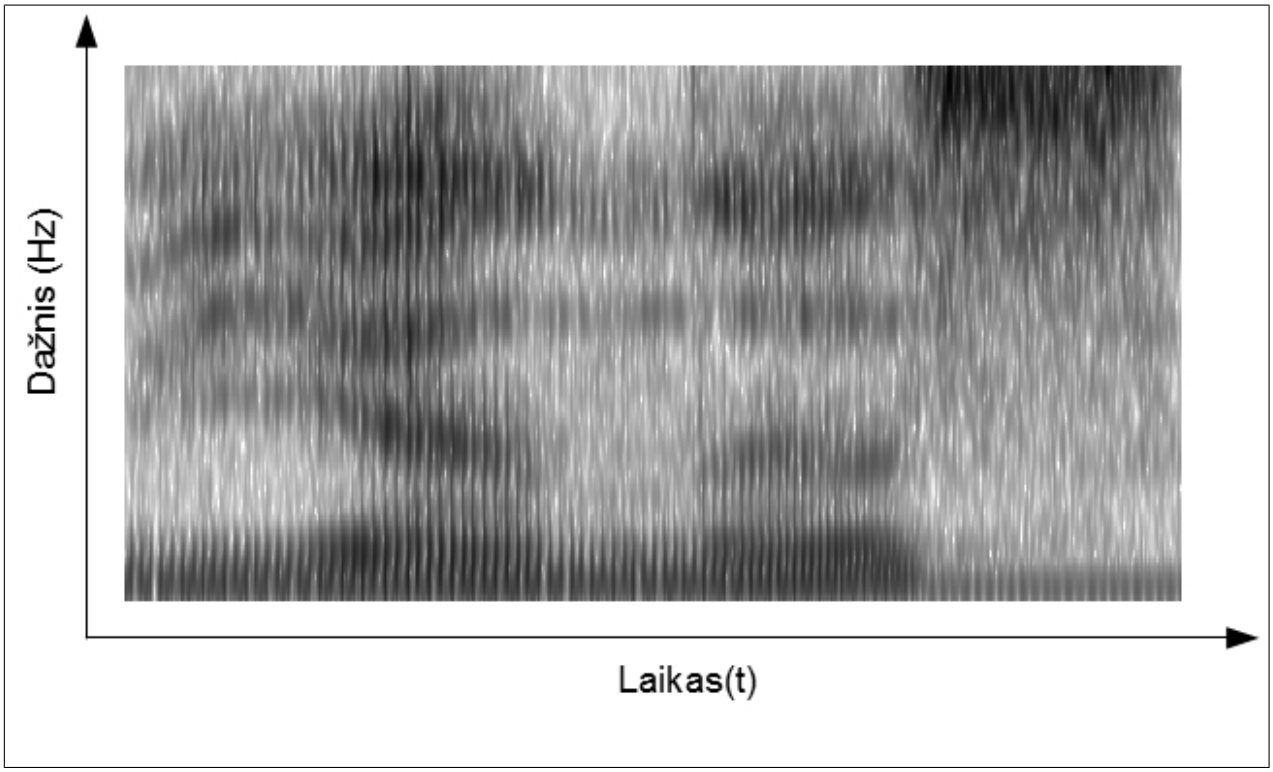
Iš paveiksliuko (1 pav. 15 psl.) matyti, kad kiekvienu vis kitu laiko momentu kinta garso slėgis, dėl ko susidaro svyravimai, kurie ir sukuria girdimą garsą.

Garso signalų dažnių spektras sudaromas remiantis garso slėgio pokyčiu dažnių atžvilgiu. Remiantis konkrečiu laiko momentu, nustatoma garso galia (slėgis) ir bangos dažnis, kuo remiantis ir sudaromas dažnių spektro grafikas (2 pav. 16 psl.). Pateikiama informacija apie bangų amplitudės kitimo intervalą ties tam tikru dažniu.



2 pav.: Ištarto žodžio „vienas“ garso signalo spektras.

Garso signalų spektograma (3 pav. 16 psl.) pateikia informacija apie garso bangos galią tam tikrame dažnių spektre. Spektogramą galima apibūdinti kaip funkciją, kurios parametrai yra dažnis ir laikas, o gaunamas rezultatas yra bangos galia(amplitudės dydis) esant tam tikram bangos dažniui ir tam tikram laiko momentui, ir šis parametras atvaizduojamas panaudojant vizualines priemones - spalvas.



3 pav.: Tariamo žodžio „vienas“ garso signalo spektrograma.

Spektrogramoms sudaryti dažniausiai naudojama diskrečioji furje transformacija (ang. DFF - Discrete Fourier Transform). Praktiniais sumetimais naudojama greitoji furje transformacija (ang. FFT - Fast Fourier Transform), kuri daug praktiškesnė skaičiavimų prasme. Kadangi tokio tipo spektrograma pateikia daugiausiai duomenų, tai dėl to ji kalbos atpažinime dažnai naudojama, išskiriant kalbos požymius.

6.1. Lango funkcija

Lango funkcijos pagrindinė paskirtis yra išryškinti signalo atkarpos centrinę dalį, o kraštų svarbą sumažinti, tokiu būdu koncentruojama informacija į centrą. Duomenys skaitomi slenkančiu kadru, kuris apibrėžiamas kadro ir kadro poslinkio dydžiais. Pavyzdžiui, jei lango dydis yra 512 kadrų, o poslinkis yra 256 kadrų, tai reiškia, kad kiekvienas langas persidengia 50 procentų, t. y. nuskaičius langą ir atlikus visus veiksmus su juo, paslenkamas langas per 256 kadrų ir po to vėl nuskaitomas langas - 512 kadrų. Paprastai naudojami Hemingo (ang. Hamming) arba Heningo (ang. Hanning) lango funkcijos.

Lango funkciją apibrėškime $w(n)$, kadrą $x_l(n)$, apdorotą kadrą $\tilde{x_l}(n)$, kur n yra kadro numeris, M – kadro dydis imtimis, o m – imtis, su kuria dirbama:

$$\tilde{x_l}(n) = x_l(n) \cdot w(n). \tag{21}$$

Hemingo langas:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(2\pi m) & \text{kai } 0 \leq m \leq M-1 \\ 0 & \text{kitu atveju.} \end{cases}, \tag{22}$$

Heningo langas:

$$w(n) = \begin{cases} 0.5 - 0.5\cos(2\pi m) & \text{kai } 0 \leq m \leq M-1 \\ 0 & \text{kitu atveju.} \end{cases}. \tag{23}$$

6.2. Tiesinės prognozės koeficientai (kodavimas)

Šis skyrius parašytas pagal [JAB09] autoriaus darbą.

Tiesinės kalbos prognozės koeficientai (kodavimas)(ang. LPC - Linear Prediction Coefficients(Coding)) - yra vienas iš galingiausių tiesinių metodų, naudojamų kalbos atpažinime. Tiesinės prognozės uždavinys gali būti suformuluotas taip: koeficientų radimas a_k, kurie išplaukia iš geriausios kalbos mėginio $s[n]$ prognozės(kurios kvadratinės prognozės klaidos vidurkis mažiausias) pagal buvusius mėginius $s[n-k], k = [1, ..., P]$. Prognozuotas mėginys $\hat{n}[n]$ gaunamas pagal

$$\hat{s}[n] = \sum_{k=1}^{P} (a_k s[n-k]), \tag{24}$$

kur P skaičius buvusių mėginių iš $s[n]$, kuriuos norima egzaminuoti.

Toliau ieškomi prognozės koeficientai a_k. Remiantis (24) formule, kai nuspėtas mėginys lygus tikram signalui(pvz.: $\hat{s}[n] = s[n]$), tai [JAB09]

$$s[n] = \sum_{k=1}^{P} (a_k s[n-k]), \tag{25}$$

$$s(z) = \sum_{k=1}^{P} (a_k s(z) z^{-k}), \tag{26}$$

$$s(z) = \frac{1}{1 - \sum_{k=1}^{P} (a_k z^{-k})}. \tag{27}$$

Optimalus sprendimas šiam uždaviniui yra [JAB09]

$$a = (a_1, a_2, \ldots, a_P),$$

$$r = (r_{ss}[1], r_{ss}[2], \ldots, r_{ss}[P])^T,$$

$$R = \begin{bmatrix} r_{ss}[0] & r_{ss}[1] & \ldots & r_{ss}[P-1] \\ r_{ss}[1] & r_{ss}[0] & \ldots & r_{ss}[P-2] \\ \vdots & \vdots & \vdots & \vdots \\ r_{ss}[P-1] & r_{ss}[P-2] & \vdots & r_{ss}[0] \end{bmatrix},$$

$$a = R^{-1} r. \tag{28}$$

Dėl matricos **R** Tioplico savybės (yra simetriška su lygiomis įstrižainės elementais) egzistuoja efektyvus algoritmas skirtas apskaičiuoti a be brangių skaičiavimų surandant R^{-1}. **Levinson – Durbin algoritmas** - metodas skirtas apskaičiuoti koeficientus a [JAB09].

Pradinis žingsnis:

$E_0 = r_{ss}[0], i = 1$

for $i = 1$ **to** P.

Žingsniai:

1. $k_i = \frac{1}{E_{i-1}} (r_{ss}[i] - \sum_{j-1}^{i-1} (a_{j,i-1} r_{ss}[|i-j|]))$
2. $a_{j,i} = a_{j,i-1} - k_i a_{i-j,i-1}, j = 1, ..., i-1, a_{i,i} = k_i$
3. $E_i = (1 - k_i^2) E_{i-1}$

6.3. Iškraipytos tiesinės prognozės koeficientai (kodavimas)

Iškraipytos tiesinės prognozės koeficientai (kodavimas) (ang. WLPC - Warped Linear Prediction Coefficients (coding)) - tai yra LPC modifikacija, kurioje spektrinės analizės atvaizdavimas yra pakeistas. Pavyzdžiui, pakeičiant LPC uždelsimo parametrą į pirmos-eilės „all-pass“ filtrą. Tai gali turėti teigiamos įtakos sumažinant bitų srautą, reikalingą garso įrašo analizei ir ypač kodavimui.

Pagrindinis pranašumas šio metodo yra tas, kad jis gali sutvarkyti dažnių skiriamąją gebą signalo spektre, kuri yra santykinai panaši į žmogaus klausą atitinkančią skiriamąja gebą. Tai padeda automatiškai panaudoti klausos charakteristikas.

Reikėtų paminėti šio metodo modifikaciją, kurią atliko šaltinio [MGR05] autoriai. WLPC-SC (ang. Warped LPC – based Spectral Centroid) yra atlikta keletas modifkiacijų tokiu kaip: dažnių skalės transformavimas į *Bark* skalę, panaudotas „one-pole“ WLPC filtras, kuris paremtas *bilinear* transformacija. Taip pat naudojama *spektrinis masių centras(ang. spectral centroid)* - tai matas, skirtas charakterizuoti skaitmeninių signalų spektrui.

Gauti rezultatai, atliekant klasifikavimą pagal tembro požymius.

FEATURE	SPEECH (%)	MUSIC (%)	GLOBAL (%)
SC	92.26	95.54	93.90
SR	95.24	90.18	92.60
SF	90.09	71.81	80.56
ZC	93.26	88.54	90.80
MFCC	92.08	99.20	95.83
LE	88.75	78.00	86.19
WLPC-SC	94.87	91.63	93.20

4 pav.: Klasifikavimo tikslumas procentais. [MGR05]

Palyginami WLPC-CS su LPC-CS.

FEATURE	SPEECH (%)	MUSIC (%)	GLOBAL (%)
WLPC-SC	94.87	91.63	93.20
LPC-SC	89.72	76.36	82.75

5 pav.: Klasifikavimo tikslumas procentais. [MGR05]

Remiantis šiomis lentelėmis, aiškiai matyti, kad kalbos klasifikavime pasirodo geriausiai WLPC, o muzikos - antroje vietoje. Atliktas palyginimas su LPC modifikacija LPC-CS parodo, kad WLPC-CS yra kur kas pranašesnis.

7. Kalbos atpažinimas

Kalbos atpažinimas (dar vadinamas **automatiniu kalbos atpažinimu** arba **kalbos atpažinimu kompiuteriu**) - konvertuoja pasakomus žodžius į tekstą. Terminas „balso atpažinimas“ kartais naudojamas nurodant, kad atpažinimo sistema turi būti apmokyta tam tikram kalbėtojui, kaip daugelyje paruoštų atpažinimo sistemų.

Kalbos atpažinimas yra sprendimas, skirtas atpažinti kalbą nesitaikant tik į vieną kalbėtoją, pavyzdžiui, sistemos valdoma balsu, kurios gali atpažinti pasirinktus balsus.

Kalbos atpažinimo aplikacijos apima „vartotojo sąsają balsu“ (ang. voice user interface), kuri leidžia atlikti tokias funkcijas kaip skambinimas balsu, skambučių peradresavimas, automatizuota namų prietaisų kontrolė, paieška, paprasta duomenų įvestis, struktūrizuoto dokumento paruošimas [wiki11a].

Šiuolaikinės kalbos atpažinimo sistemos naudoja *akustinį modeliavimą* ir *kalbos modeliavimą*, kurios yra sudedamoji dalis modernių statistika paremtų kalbos atpažinimo algoritmų. Paslėptieji Markov'o modeliai (HMMs) yra plačiai naudojami daugelyje sistemų.

Remiantis tekstu [Pik06], kalbos atpažinimo sistemos bendru atveju yra klasifikuojamos į diskrečias ir ištisines sistemas, kurios yra priklausomos, nepriklausomos nuo kalbėtojo arba adaptyvios. Diskrečios sistemos dirba su atskirais kiekvieno žodžio akustiniais modeliais, žodžių kombinacijomis ar frazėmis ir tai vadinama izoliuotos kalbos atpažinimu. Ištisinės, vientisos kalbos atpažinimo sistemos, viena vertus, reaguoja į vartotoją, kuris taria žodžius, frazes, sakinius, kurie yra iš eilės išsidėstę ar tam tikra tvarka priklausomi vienas nuo kito, taip lyg būtų susieti kartu.

Sistemos, priklausomos nuo kalbėtojo, reikalauja, kad vartotojas įrašytų žodžio, sakinio ar frazės pavyzdį prieš leidžiant sistemai atpažinti jį. Vartotojas šitaip „apmoko“ sistemą. Nuo kalbėtojo nepriklausomos sistemos kuriamos taip, kad galėtų veikti su bet kokiu, tam tikra kalba kalbančiu kalbėtoju (pvz.: lietuvių k.). Adaptyvi sistema kalbėtojo atžvilgiu yra kuriama norint pritaikyti sistemos operacijų charakteristikas prie naujo vartotojo.

Izoliuotos kalbos atpažinimo sistemos - šiuo metu yra žymiai lengvesnė užduotis kompiuteriams nei atlikti vientisos kalbos atpažinimą. Pačios moderniausios šiuolaikinės kalbos atpažinimo sistemos naudoja tikimybinius modelius, norint interpretuoti garsų seką. „Hidden Markov Model“ (HMM) modelis ypač naudojamas atpažinti žodžius. Norint pagerinti žodžių atpažinimo tikslumą, naudojamas kalbos modelis, norint pagauti informaciją, kuri parodytų, kad tam tikro žodžio kombinacijos su vienais žodžiais yra labiau tikėtinos nei su kitais.

Automatinis kalbos atpažinimas vykdomas prastai triukšmingoje aplinkoje, ypatingai kai kalba keli žmonės ar kai kas nors trukdo pokalbį. Žmonės triukšmingoms aplinkoms yra labai tolerantiški, o kompiuteriai triukšmą toleruoją sunkiai, dėl to automatinio kalbos atpažinimo kokybė ir tikslumas mažėja drastiškai, kai triukšmo lygis didėja. Garso signalų iškraipymai nuo foninių pokalbių aplinkoje, kurioje yra daug šnekančių žmonių, yra ypatingai sudėtingi. Dirbtiniai neuronų tinklai teikia nemažai vilčių, sprendžiant tokio tipo problemas.

Svarbus dalykas kuriant kalbos atpažinimo sistema yra žodyno dydis. Kalbos atpaži-

nimo žodyno dydis paveikia sudėtingumą, apdorojimo reikalavimus ir sistemos tikslumą. Akivaizdu, jog daug lengviau ir paprasčiau atlikti vieno žodžio paiešką žodyne, kuriame yra 20 žodžių, nei žodyne, kurį sudaro šimtas tūkstančių žodžių. Tai yra vienas iš esminių darbų, ką daro kalbos atpažinimo sistema.

Kitas svarbus rodiklis yra kalbos tipas, kurį naudoja atpažinimo sistema: diskreti ar vientisa kalba. Diskrečioje kalbos sistemoje vartotojas privalo padaryti pauzę tarp kiekvieno žodžio, kas padaro kalbos atpažinimo uždavinį daug lengvesnį. Šį atpažinimo metodą paprasčiausia atlikti, nes žodžio pradžias ir pabaigas lengviau surasti ir tarimas nepaveikia taip kitų žodžių. Kadangi žodžių vartojimas yra pastovus, tai ir palengviną darbą.

Vientisos kalbos atpažinimo sistemos operuoja kalba, kurią sudaro tarpusavyje sujungti žodžiai, t. y. neatskirti pauzėmis. Vientisą kalbą apdoroti yra sunkiau dėl didesnio kiekio efektų. Pirma, sunku surasti žodžių pradžias ir pabaigas. Antra problema yra „koartikuliacija“. Kiekvieno garso (fonemos) tarimas yra paveikiamas supančių fonemų ir dėl žodžių pradžių ir pabaigų panašumų žodžiai yra paveikiami pirmiau ar vėliau einančių žodžių. Nepertraukiamos kalbos atpažinimas yra taip pat paveikiamas kalbėjimo greičio.

Žodynas apibrėžia akustinius modelius individualiems kalbos garsams visiems naudojamiems žodžiams. Reiktų nepamiršti, kad žodynas gali turėti keletą tarimų tam pačiam žodžiui.

Akustinių modelių aibė gali būti „apmokyta“ daugelio vartotojų kalbos įrašais. Šie modeliai apima įvairias tarimo variacijas (tarmė), akcentas ir t. t. individualiai kiekvienam kalbėtojui. Svarbu „apmokyti“ modelių aibę aplinkoje, panašioje į tą, kurioje bus naudojamas kalbos atpažinimas, pavyzdžiui, reiktų netreniruoti modelių aibės naudojant ausinių mikrofono, jeigu atpažinimo aplinka įtraukia naudojimąsi telefonu.

Labai paprastas kalbos atpažinimo procesas yra tada, kai naudojami gryni (ang. raw) akustiniai duomenys. Atpažinimo sistema lygina akustinius duomenis su akustiniais modeliais naudodamas dekoderį, kuris generuoja atpažinimo hipotezes.

Kalbos atpažinimo procesas prasideda su skaitmenizuota vartotojo žodžių įvestimi. Įvestis gali būti tiesiogiai iš šaltinio, arba gali būti įrašyta.

Kitas etapas yra garso signalų apdorojimas, kur skaitmenizuoti žodžių garsai yra padalinami į seriją diskrečių „stebėjimo duomenų“ (ang. observations). Čia tikimasi, kad šie stebėjimo duomenys yra tikslūs atitikmenys vartotojo žodžio garsams.

Tolimesnis etapas - mėginimas sulyginti diskrečius stebėjimo duomenis su žinoma aibe garso modelių. Kiekvienas modelis reprezentuoja fonemą. Modelių aibė yra kombinuojama į žodį ar frazę, naudojant žodyną. Žodynas apibrėžia tarimą kiekvienam žodžiui ar frazei. Šis žingsnis gali būti atliktas naudojantis įvairiomis technikomis, tokiomis kaip „Hidden Markov Model (HMM)“, „Dynamic Time Warping (DTW)“, „Neural Networks (NNs)“, kitomis sistemomis arba šių technikų kombinacijomis. HMM - paremtos sistemos šiuo metu yra dažniausiai ir sėkmingiausiai naudojamos [Pik06].

Atliekant atpažinimo etapą, egzistuojantys treniruoti garso modeliai yra lyginami su apdorota balso įvestimi (diskretizuojami duomenys). Dekoderis (pvz. Viterabi, Baum-Welch)

yra naudojamas surasti atitikmenį balso įvesčiai pagal egzistuojančius garso modelius. Dekoderis perrašo ištisos kalbos įvestį į tekstinių simbolių seką, kurią aplikacija gali tiesiogiai apdoroti. Tikslas yra rasti atpažįstamą simbolių grupę lyginant ją su garso kalbos modeliais. Paskutinis žingsnis yra rezultatų aproksimacija ir pateikimas vartotojui [Pik06].

7.1. Akustinis modeliavimas (ang. Acoustic modeling)

Kalbos akustinis modeliavimas paprastai reiškia procesą, pateikiantį statistinę reprezentaciją požymių sekai apskaičiuotai pagal kalbos bangos formą. Paslėptasis Markov'o modelis (HMM) yra vienas iš dažniausiai naudojamų akustinių modelių. Kiti akustiniai modeliai apima segmentavimo modelius, dirbtinius neuronų tinklus, maksimalios entropijos modelius ir t. t. [MicRea].

Akustinis modeliavimas taip pat apima „tarties modeliavimą“, kuris aprašo, kaip fonemų sekos yra naudojamos sukurti didesnius kalbos vienetus, tokius kaip žodžiai ar frazės, kurie ir yra kalbos objektai, kurie yra atpažįstami. Akustinis modelis gali taip pat įtraukti grįžtamojo ryšio informacijos panaudojimą iš atpažinimo sistemos, kad prireikus galima būtų atnaujinti ar pakeisti kalbos požymių vektorių mažinant triukšmo daromą įtaką [MicRea].

Kalbos atpažinimo sistemos dažniausiai reikalauja dviejų esminių komponentų. Vienas komponentas yra akustinis modelis, sukuriamas paimant kalbos audio įrašus ir jų transkripcijas ir pagaminant iš to statistinę žodžių reprezentaciją. Kitas komponentas yra kalbos modelis, kuris pateikia žodžių sekoms tikimybes [MicRea].

7.2. Paslėptasis Markov'o Modelis

Paslėptasis Markov'o Modelis (ang. Hidden Markov Model (HMM)) metodas kalbos atpažinimo srityje naudojamas sėkmingiausiai šiuo metu . Taip pat neblogą rezultatą rodo ir dirbtiniai neuronų tinklai, tačiau šiame skyriuje bus apžvelgtas tik *paslėptasis Markov'o modelis*. Kadangi jis dažniausiai naudojamas gaunant aukštus rezultatus, todėl ir įdomus šio metodo veikimas.

Realaus pasaulio procesai bendru atveju duoda matomą rezultatą, išvestį, kurią galima apibūdinti kaip signalą. Signalas gali būti diskretus (pvz.: simboliai iš abėcėlės) arba nepertraukiami, tolydūs (pvz.: kalbos pavyzdžiai, temperatūros matavimai, muzika, ...). Signalų šaltinis gali būti pastovus arba nepastovus. Signalai gali būti sveiki, sugadinti ar iškraipyti persiunčiant ir pan.

Garsinius signalus apdorojus, gaunami požymiai, kurie naudojami modeliuojant akustinių kalbos modelį. Šiam darbui dažnai nudojamas *paslėptatis Markov'o modelis*

Paslėptasis Markov'o modelis susideda iš dviejų daliu:

- Markov'o modelio (dar vadinama Markov'o sistema, Markov'o procesu).

- Markov'o paslėptojo sluoksnio.

Markov'o sistemą (arba procesą) sudaro N būsenų $S \in \{s_1, s_2,...,s_N\}$, kur $N \in \mathbb{N}$. Diskretūs laiko momentai (žingsniai) $t = 1, t = 2,...,$ t. y. $t \in \mathbb{N}$. Kiekvienu laiko momentu Markov'o sistema yra vienoje iš galimų būsenų $q_t \in \{s_1, s_2,...,s_N\}$. Esama būsena S nulemia kitą būseną. Esant q_t būsenos pokytis q_{t+1} yra sąlygiškai nepriklausomas nuo $\{q_{t-1}, q_{t-2}, ..., q_1, q_0\}$, kitaip sakant, kita būsena q_{t+1} priklauso tik nuo vienos buvusios būsenos q_t. Aiškumo dėlei parodoma,

$$P[q_{t+1} = s_j | q_t = s_i] = P[q_{t+1} = s_j | q_t = s_i, q_{t-1} = s_k,...],$$

kad visos praeitų būsenų tikimybės atmetamos, ir paliekama paskutinė. Šis modelis tiesiogiai atspindi stebėjimo duomenis. Papildžius šį modelį būsenų tikimybine funkcija, gaunamas taip vadinamas *paslėptasis Markov'o modelis*. Palėpasis Marko'vo modelis - yra dvigubas stochastinis (tikimybinis) procesas, kuris paremtas kitu tikimybiniu procesu, kuris nėra matomas (paslėptas), bet gali būti matomas (naudojamas) tik per kitą stochastinių procesų aibę, kuri sudaroma pagal stebėjimų seką [Rab89].

Paslėptasis Markov'o modelis charakterizuojamas šiais kriterijais [Rab89]:

1. N - modelio būsenų kiekis. Nors būsenos yra „paslėptos“, tačiau daugelyje praktinių aplikacijų būsenoms yra priskiriamos realios fizinės reikšmės. Bendrai, būsenos tarpusavyje yra susijungusios viena su kita tokiu būdu, kad bet kuri būsena gali būti pasiekiama iš bet kurios kitos būsenos. Individualios būsenos apibrėžiamos taip: $S \in \{s_1, s_2,...,s_N\}$, ir būsena laiko momentu t kaip q_t.

2. M - skirtingų stebėjimo simbolių vienai būsenai skaičius, pavyzdžiui, diskrečios abėcėlės dydis. Stebėjimo duomenys atitinka sumodeliuotos sistemos fizinę išvestį. Individualūs simboliai apibrėžiami taip: $V = \{v_1, v_2, ..., v_m\}$.

3. Būsenų perėjimo tikimybių pasiskirstymas $A = \{a_{ij}\}$, kur

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N.$$

4. Stebimo simbolio tikimybės pasiskirstymas būsenoje j, $B = \{b_j(k)\}$, kur

$$b_j(k) = P\{v_k \text{ laiko momentu t} | q_t = S_j\},$$

$$1 \leq j \leq N, 1 \leq k \leq M.$$

5. Pradinių būsenų pasiskirstymas $\pi = \{\pi_i\}$, kur

$$\pi_i = P\{q_1 = S_i\}, \qquad 1 \leq i \leq N.$$

Su N, M, A, B, π tinkamomis reikšmėmis, HMM gali būti naudojamas kaip generato-

rius, kuris duotų stebimų duomenų seką

$$O = O_1 O_2 ... O_T,$$

(kur kiekvienas parametras O_t yra vienas simbolis iš V, ir T yra duomenų skaičius sekoje) generuoja tokia seka:

(a) Pasirenkama pradinė būsena $q_1 = S_i$ remiantis pradinių būsenų pasiskirstymu π.

(b) Nustatomas $t = 0$.

(c) Pasirenkamas $O_t = v_k$ remiantis simbolio tikimybiniu pasiskirstymu būsenoje S_i, pavyzdžiui, $bi(k)$.

(d) Pereinama į naują būseną $q_{t+1} = S_j$ remiantis būsenų perėjimo tikimybinių pasiskirstymu būsenai S_i, pavyzdžiui, a_{ij}.

(e) Nustatomas $t = t + 1$; grįžtama į (c) žingsnį, jeigu $t < T$; priešingu atveju nutraukiama procedūra.

Aukščiau aprašyta procedūra, gali būti naudojama ir stebėjimo duomenų generavimui, ir kaip modeliu, mėginant surasti sąryšius pagal stebėjimo duomenis.

HMM reikalingi du modelio parametrai (N ir M), stebėjimo simboliai ir trys tikimybių matai A, B ir π. Paprastumo dėlei aprašykime tai kompaktiškai

$$\lambda = (A, B, \pi)$$

pažymint modelio pilną parametrų aibę.

7.3. Kalbos modeliavimas (ang. Language modeling)

Kalbos modeliavimo (dar vadinama statistiniu kalbos modeliavimu) pagrindinis tikslas yra sukurti statistinį kalbos modelį, kuris galėtų kaip galėdamas tiksliau įvertinti kalbos išsidėstymą. Statistinis kalbos modelis - tai tikimybių išsibarstymas tekste, kuris mėgina atspindėti, kaip dažnai žodis ar frazė atrandama kaip sakinys [Zha10].

Išreiškiant įvairius kalbos požymius laiko atžvilgiu kaip parametrus statistiniame modelyje, statistinis kalbos modelis suteikia lengvą būdą susitvarkyti su sudėtinga kalba kompiuteryje.

Paprastai kalbant, tokie modeliai naudojami pagauti kalbos požymius ir nuspėti kitą žodį kalbos sekoje. Vienas iš tokių algoritmų yra *N-gram*. Tai tikimybinis kalbos modelis, kuris pateikia labiausiai naudojamą simbolių eilutę (žodį). Kalbos modeliai taip pat gali būti panaudojami, norint nuspėti kaip sakinys užsibaigs, t. y. gali pateikti spėjimą, kas yra rašoma ir pateikti pasiūlymą, kaip užbaigti sakinį.

7.4. Kalbos atpažinimo schema

Kalbos atpažinimas susideda iš kelių pagrindinių dalių. Kaip matyti iš schemos (6 pav. 26 psl.) visas kalbos atpažinimo procesas susideda iš 9 dalių, nors minimaliai užtenka ir pirmų 7 dalių.

1. *(1 -> 2)* - žingsnyje žodis ar frazė yra pasakoma į mikrofoną. Garso korta analoginį garso signalą skaitmenizuoja ir padaro „suprantamą“ kompiuteriui.

2. *(2 -> 3)* fazėje vyksta akustinis garso apdorojimas. Signalas yra diskretizuojamas (padalinamas į mažesnes dalis).

3. *(3 -> 4)* atliekamas požymių išskyrimas naudojant MFCC (mel-frequency cepstral coefficients), PLP (perceptually linear prediction coefficients), LPC(Linear Prediction Coefficients) ir pan.

4. *(4 -> 5)* išgauti įrašai paduodami dekoderiui. Dekoderį sudaro *akustinis kalbos modelis* ir *kalbos modelis* bei paieškos algoritmas, skirtas surasti duotai frazei atitikmenį.

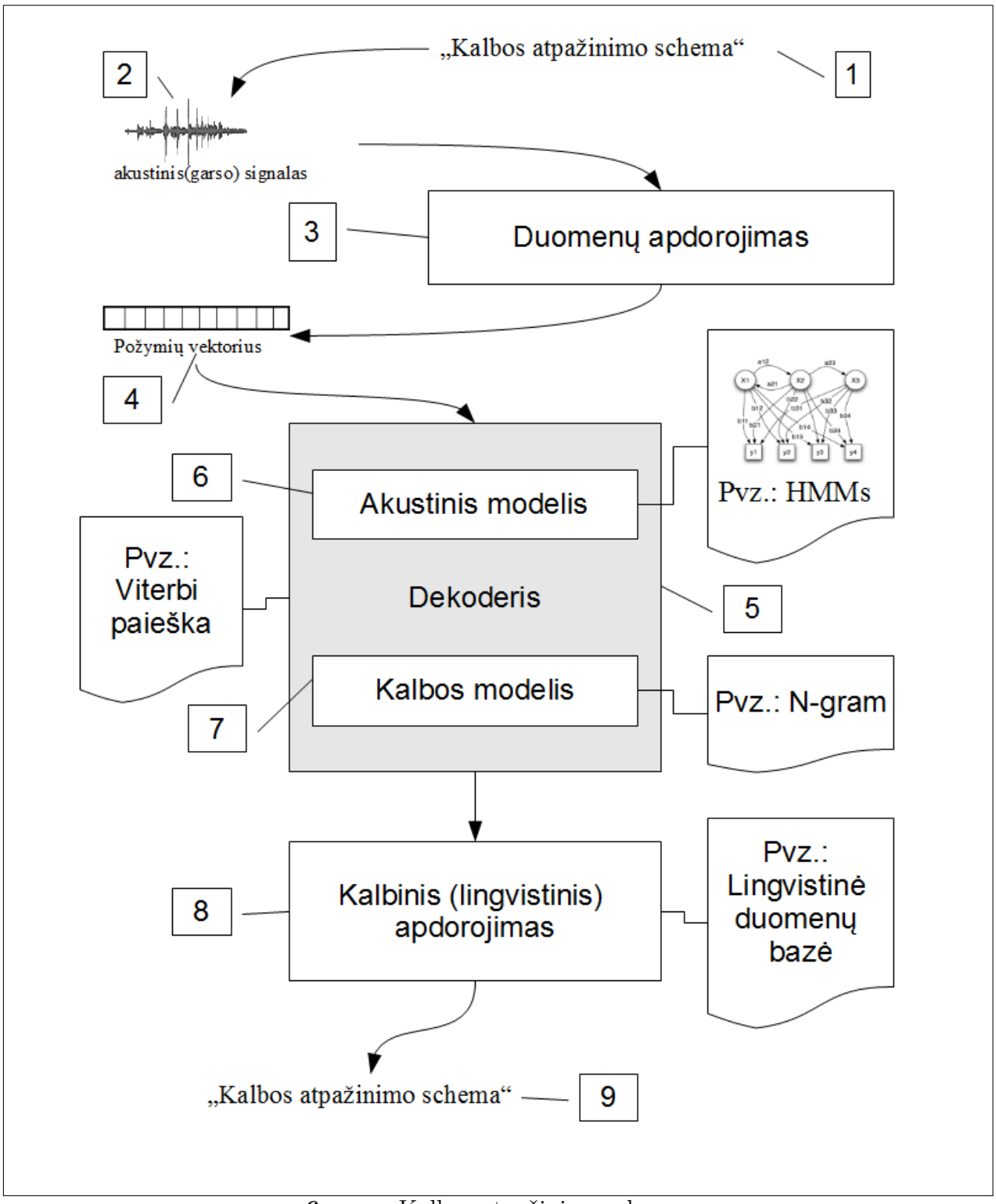
5. *(5 -> 6)* vykdomas akustinis modeliavimas. Šiam darbui dažniausia naudojami Hidden Markoc Models (HMMs) ir Artificial Neural Networks(ANNs).

6. *(6 -> 7)* vykdomas kalbos modeliavimas. Tam tikslui galima naudoti *N-gram* algoritmą.

7. **(7 -> 9)* Po šio proceso jau turime kalbos atpažinimo sistema. Tačiau dauguma šiuolaikinių sistemų dar naudoja papildomus metodus, pavyzdžiui, lingvistines duomenų bazes, kurios užtikrina rezultatų teisingumą.

8. *(7 -> 8)* atliekamas kalbinis (lingvistinis) apdorojimas.

9. *(8 -> 9)* gautų rezultatų pateikimas.



6 pav.: Kalbos atpažinimo schema.

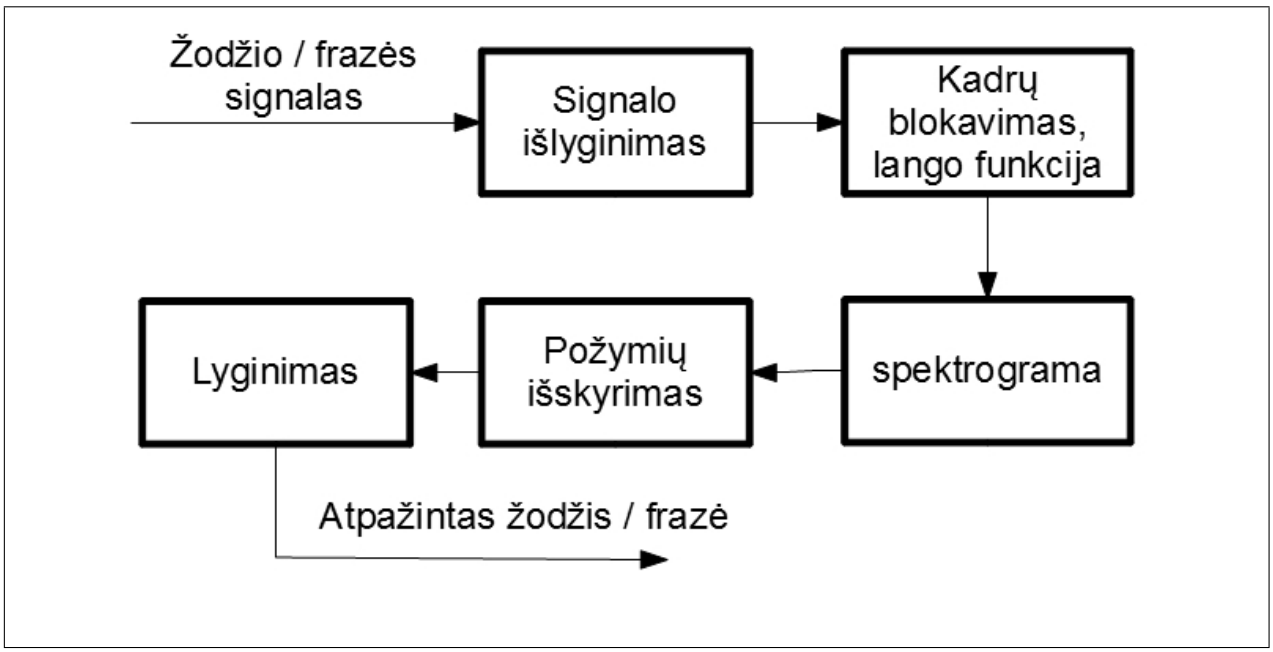
8. Kalbos ir asmens atpažinimas panaudojant spektrogramą ir jos požymius

Šiame skyrelyje aprašoma sukurtos kalbos atpažinimo sistema. Pateikiami pažingsniui visi veiksmai, kurie trumpai aprašomi, o kur reikalingas platesnis paaiškinimas, pateikiama tolimesniame tekste.

8.1. Kalbos atpažinimo sistemos dizainas

Bendra kalbos atpažinimo schema susideda iš kelių blokų:

1. Įvestis.
2. Signalo išankstinis apdorojimas.
3. Signalo suskaidymas į kadrus ir jų apdorojimas lango funkcija.
4. Spektrogramos sudarymas.
5. Požymių išskyrimas.
6. Lyginimas.
7. Išvestis.



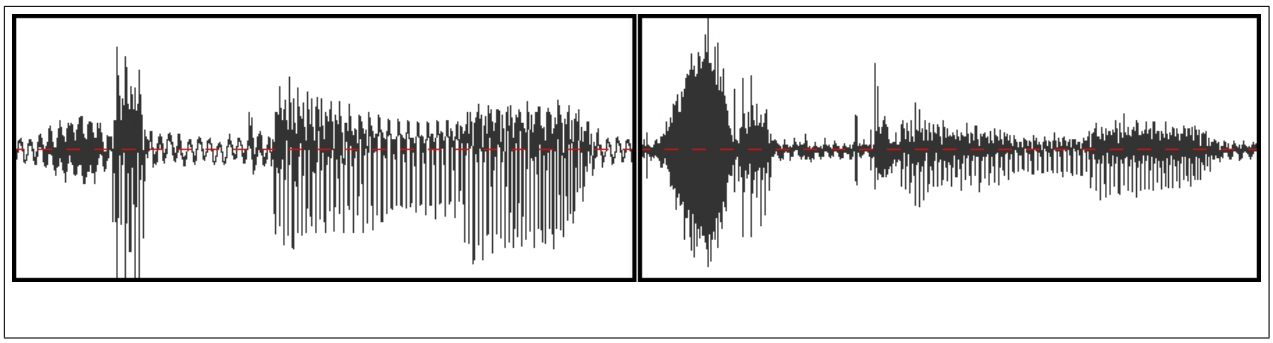
7 pav.: Kalbos atpažinimo schema.

1. *Spektro išlyginimas (ang. Pre-emphasis):* Skaitmeninis kalbos signalas $s(n)$ praleidžiamas pro žemos eilės skaitmeninę sistemą, tam, kad spektriškai sulygintų signalą ir sumažintų jautrumą tikslumui vėlesniuose signalo apdorojimo žingsniuose. Taip pat

šis metodas padeda kai kuriems dažniams atitrūkti nuo triukšmo. Išvestis gaunama apdorojus signalą funkcija:

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1). \tag{29}$$

Dažniausiai $\tilde{a}$ reikšmė yra apie 0.95.



8 pav.: Kairėje originalus signalas, dešinėje apdorotas „Pre-emphasis“ filtru. Naudota $\tilde{a} = 0{,}9375$.

2. *Kadrų blokavimas (ang. Frame Blocking):* Išvestis, atlikus pirminio signalo iškraipymą, $\tilde{s}(n)$, yra suskirstomas į kadrus iš N mėginių (ang. samples), o kaimyniniai kadrai atskirti per M mėginių. Jeigu x_l yra l – is signalo kadras ir visame kalbos signale yra L kadrų, tai

$$x_l(n) = \tilde{s}(Ml + n), \tag{30}$$

kur $n = 0{,}1, \ldots, N-1$ ir $l = 0{,}1, \ldots, L-1$.

3. *Lango funkcijos panaudojimas (ang. Windowing):* Po kadrų blokavimo, kitas žingsnis yra lango funkcijos pritaikymas kiekvienam kadrui, taip sumažinant signalo nutrūkimus kadro pradžioje ir pabaigoje. Jeigu mes lango funkciją apibrėšime kaip $w(n), 0 \leq n \leq N-1$, tai rezultatas pritaikius lango funkciją yra:

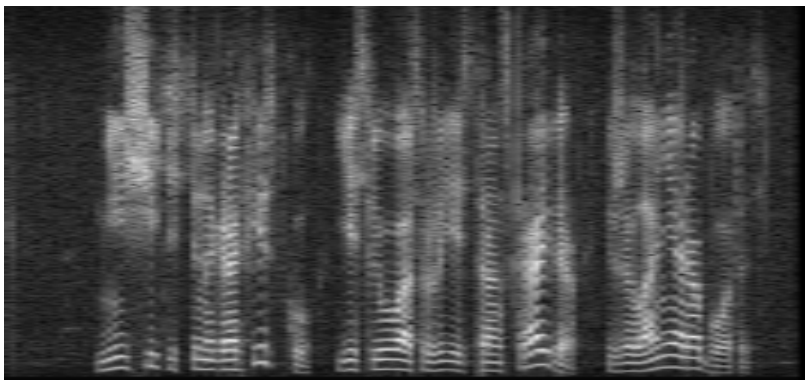
$$\tilde{x}_l(n) = x_l(n)w(n), \tag{31}$$

kur $0 \leq n \leq N-1$.

4. *Spektrogramos sudarymas:* Sudaroma audio įrašo spektrograma, kuria remiantis išskiraimi įrašo požymiai. Spektrogramos sudarymui naudojam Furje transformacija, kuri įrašą konvertuoją iš laiko srities į dažnių sritį. Kiekvienas kadras apdorotas lango funkcija paduodamas Furje funkcijai:

$$E = e_l(n) = dft(\tilde{x}_l(n)), \tag{32}$$

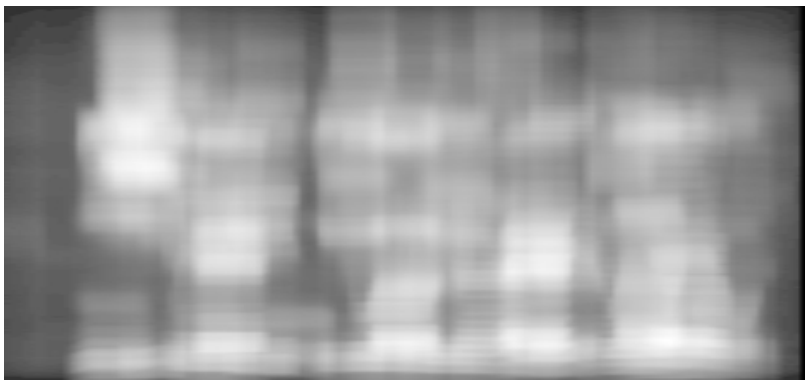
E - audio įrašo spektrograma.



9 pav.: Sugeneruota spektrograma.

5. *Spektrogramos vidurkinimas:* Vidurkinant spektrogramą atsiribojama nuo konkretaus žmogaus. Vidurkinimas atliekamas x ir y kryptimi.

$$\tilde{E} = mean(E, xRadius, yRadius). \tag{33}$$



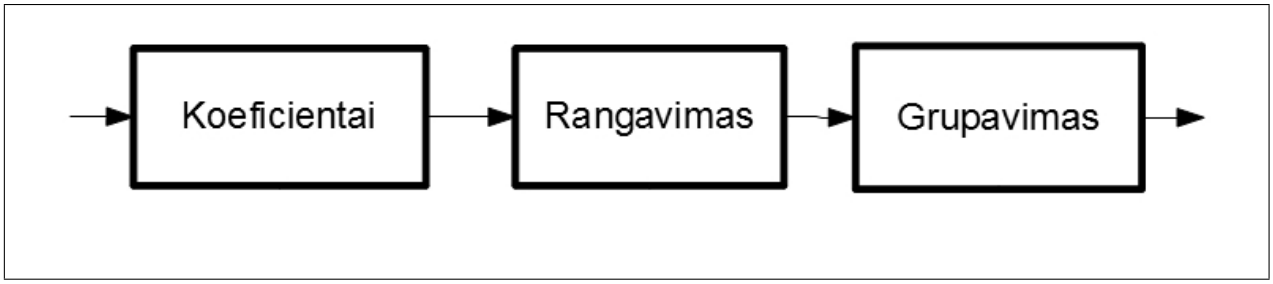
10 pav.: Vidurkinta spektrograma.

6. *Požymių išskyrimas:* Įrašo požymiai yra esminiai duomenys, kuriais remiantis atliekamas atpažinimas. Iš suvidurkintos spektrogramos išrenkami kas antri stulpelio taškai ir iš jų apskaičiuojami Teiloro eilutės koeficientai. Jie ranguojami ir gaunami binarizuoti duomenys, dar sugrupuojami ir turime galutinius įrašo požymius. Plačiau aprašytas šis žingsnis skyriuje „Požymių išskyrimas“.

7. *Lyginimas (ang. Matching):* Tai procesas, kurio metu lyginamas realus įrašas su etalonų duomenų aibę, pagal kurią surandamas geriausias atitikmuo. Lyginimas vyksta naudojant tik požymius, t. y. lyginami požymiai iš etalonu duomenų aibės ir lyginamo įrašo požymiai. Gautą rezultatų aibę normuojame, kad gautume geresnė metriką t. y. tikslesnį atpažinimą. Skyriuje "Požymių lyginimas"pateikta išsamiau.

9. Požymių išskyrimas

Požymių išskyrimas atliekamas keletu etapų. Pirmo etapo metu surandami Teiloro eilutės koeficientai iš suvidurkintos įrašo spektrogramos. Antru etapu koeficientai yra ranguojami ir po šio veiksmo gaunami binarizuoti duomenys, t. y. duomenys, sudaryti iš 1 arba 0. Trečiu žingsniu požymiai sugrupuojami.



11 pav.: Požymių išskyrimo schema.

Kadangi binarizaciją iš esmės sudaro Teiloro koeficientų radimas ir rangavimas, tai viskas apjungta skyrelyje „Binarizavimas“, o grupavimas pateiktas kaip atskiras skyrelis.

9.1. Binarizavimas

Sugeneruojama pasirinkto aukščio spektrograma. Tarkim pasirenkam spektrogramos aukštį $N = 256$. Jei jūsų Furje transformacija grąžina 1024 reiškmes, tai puse jų atmetame dėl simetriškumo, toliau vidurkinama ir po vidurkinimo praretiname spektrogramą, t. y. binarinius požymius sudarinėjame kas antram spektrogramos stulpelio taškeliui. Pažymėkime $u(f,t)$ kaip išretintą spektrogramą, kur $f = 0, 1, 2, ..., N$, $t = 0, 1, ..., T-1$, T – pilnas kadrų skaičius.

Binarizacija atliekama keturiomis matricomis. $v_k(f,t)$, $k = 0, 1, 2, 3$. Šios matricos imituoja pirmuosius keturis Teiloro koeficientus:

1. $v_0(f,t) = u(f,t)$, t. y. imame spektrogramą $u(f,t)$.
2. $v_1(f,t) = u(f,t+1) - u(f,t-1) = v_0(f,t+1) - v_0(f,t-1)$ - imituojama spektrogramos išvestinė laike.
3. $v_2(f,t) = v_1(f,t+2) - v_1(f,t-2)$ - imituojama spektrogramos antros eilės išvestinė.
4. $v_3(f,t) = v_2(f,t+3) - v_1(f,t-3)$ - imituojama spektrogramos trečios eilės išvestinė.

Kiekviena matricą $v_k(f,t)$ binarizuojama atskirai, t. y. $v_k(f,t)$ reikšmės paverčiamos į 0 arba 1.

Fiksuojama eilutė $v_k(f,\cdot)$, turime T eilės vektorių. Taip pat fiksuokime slenkantį langą, kurio dydis $-50, -49, \ldots, 0, 1, \ldots, 49, 50$.

Pasirenkame kokį nors nors eilutės taškelį

$$p_{v_k} = v_k(t,f), \tag{34}$$

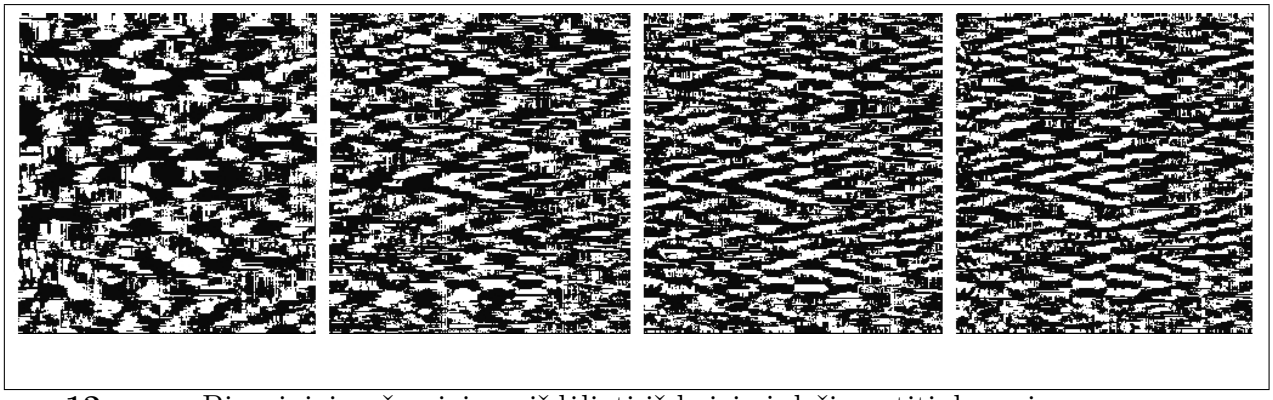
ir lango rėmuose surandame kiek už p_{v_k} yra didesnių reikšmių. Jei paslinktos t indekso reikšmės išeina iš ribų, tokiems t elementų iš vis nebinarizuojame. Suskaičiavę lango ribose kiek yra didesnių reikšmių už p_{v_k}, gaunama $rankRows(f,t)$ matrica, kurioje yra sveikos reikšmės nuo 0 iki 100.

Atliekame analogišką procedūrą stulpeliui su $rankRows(f,t)$ matrica. Fiksuojama stulpelį $rankRows(\cdot,t)$ ir lango ribose nuo -32 iki 32 skaičiuojama kiek už

$$p_{rankRows} = RankRows(f,t) \tag{35}$$

reikšmę yra lango ribose didesnių reikšmių. Suskaičiavus gaunama nuo 0 iki 64. Jei reikšmė mažiau už 32 ($p_{rankRows} < 32$), priskiriamas binarinis požymis $rank(f,t) = 0$, jei didesnis už 32 ($p_{rankRows} > 32$), priskiriama $rank(f,t) = 1$. Jei lygiai lygu 32, tuomet pasižiūrima ar $v_k(f+1,t) > v_k(f,t)$. Jei taip, $rank(f,t) = 0$, kitu atveju $rank(f,t) = 1$.

Išėjus indeksams iš matricos ribų, naudokite periodinį pratęsimą ($v_k(N,t) = v_k(0,t)$ ir t. t.), nes spektro kryptimi jis natūralus.

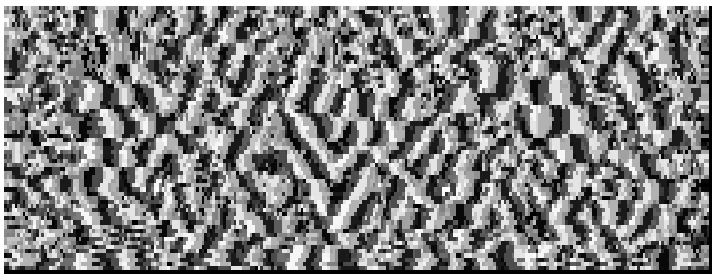


12 pav.: Binariniai požymiai, v_k išdėlioti iš kairės į dešine atitinkamai v_0, v_1, v_2, v_3.

9.2. Grupavimas

Pirmiausiai sugrupuojame požymius dėl patogumo ir dėl lyginimo spartos padidėjimo. Požymių masyvų $v_k(f,t)$ elementai sugrupuojami taip, kad butų išsidėste paeiliui t. y.

$$\begin{aligned} g(t,c) = &[[v_0(0,0), v_1(0,0), v_2(0,0), v_3(0,0)], \\ &[v_0(0,1), v_1(0,1), v_2(0,1), v_3(0,1)], \\ &\cdots, \\ &[v_0(1,0), v_1(1,0), v_2(1,0), v_3(1,0)], \\ &\cdots, \\ &[v_0(N,T), v_1(N,T), v_2(N,T), v_3(N,T)]]. \end{aligned} \tag{36}$$



13 pav.: Sugrupuoti binariniai požymiai.

10. Požymių lyginimas

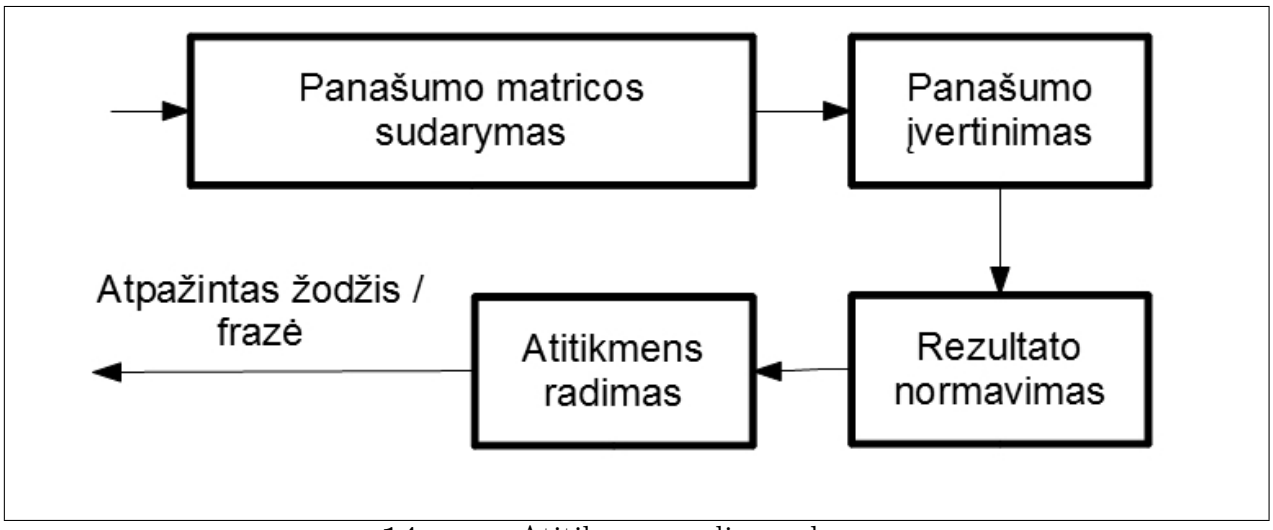
Įrašai lyginami su etalonų duomenų baze, pagal kuria atpažįstamos frazės. Kiekvienas įrašas prieš palyginimą yra apdorojamas t. y. išskiriami požymiai, kuriais remiantis įrašai lyginami. Lyginimas susideda iš keleto etapų:

1. Pirmas etapas, kurio metu sudaroma panašumo matrica (kaupiamasis panašumas), kurioje lyginami įrašų kadrų.

2. Antras etapas, kurio metu naudojantis DTW algoritmų įvertinamas panašumas.

3. Trečias etapas, rezultato normavimas. Normuotas rezultatas pagerina atpažinimo metriką.

4. Išrenkamas atitikmuo su geriausiu įverčiu.



14 pav.: Atitikmens radimo schema.

10.1. Kaupiamasis panašumas

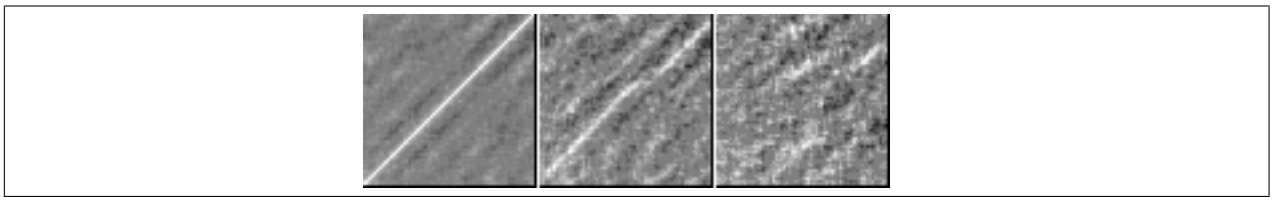
Kaupiamasis panašumas pateikia įvertinimą kaip panašūs lyginamų kūrinių kadrai. Kaupiamasis panašumas pateikiamas matricos forma, tai pažymėkime kaip $c(i, j)$. Pažymėkime $A(t, a)$ – A įrašo požymių masyvas, o $B(t, b)$ – B įrašo požymių masyvas, a, b – atitinkamai sugrupuotų požymių skaičius masyve, z_i^A – A įrašo i – is požymių kadras, z_j^B atitinkamai B įrašo j – is požymių kadras.

Atliekant dviejų įrašų lyginimą, lyginimo pagrindas tampa kadrų požymių panašumo lyginimas. Kadrai lyginami vienas prie vieno, t. y. lyginami $z_i^A(n)$ su $z_j^B(n)$, n – kadro elemento numeris. Kadangi požymiai binarizuoti, tai pats lyginimo procesas labai paprastas. Kadrų panašumas apskaičiuojamas atliekant paprasta loginę XOR operaciją

$$\lambda_{(i,j)}^{(A,B)} = \sum_{n=0}^{T} z_i^A(n) \oplus z_j^B(n), \tag{37}$$

kur l – įrašo kadro numeris. Toliau apskaičiuotas kadrų panašumas naudojamas sukurti kaupiamojo panašumo masyvą

$$c(i,j) = \lambda_{(i,j)}^{(A,B)}. \tag{38}$$



15 pav.: Panašumo matricos. Kairėje yra idealus atitikmuo etalonų įrašui, viduryje - ta pati frazė sakoma to pačio žmogaus kitu laiko momentu, dešinėje - kita frazė, sakoma to pačio žmogaus.

10.2. DTW algoritmas

DTW (ang. Dynamic Time Warping) – algoritmas, skirtas matuoti panašumą tarp dviejų, laike kintančių, duomenų sekų. Kadangi žmogaus kalba yra signalas, kintantis laike, tai šių duomenų palyginimui šis algoritmas tinkamas.

Apibrėžkime DTW algoritmo rezultatų matricą kaip $M_{dtw}(N, M)$, kur N – eilučių, o M – stulpelių skaičius. Atstumo įverčio funkciją $\Delta_{dtw}(f_a, f_b)$, kuri apskaičiuoja panašumo įvertį tarp dviejų reikšmių ir tam darbui gali būti naudojamas Euklidinis atstumas ar Kosinuso panašumo įvertis ar kita atstumo įverčio sistema.

Parametrų pradiniai nustatymai nustatomi priklausomai nuo sąlygų:

1. Jeigu DTW algoritmas minimizuoja reikšmes tai:

$$\begin{aligned} M_{dtw}(0,0) &= 0, \\ M_{dtw}(0,\cdot) &= \infty, \\ M_{dtw}(\cdot,0) &= \infty. \end{aligned} \tag{39}$$

2. Jeigu DTW algoritmas maksimizuoja reikšmes tai:

$$\begin{aligned} M_{dtw}(0,0) &= 0, \\ M_{dtw}(0,\cdot) &= -\infty, \\ M_{dtw}(\cdot,0) &= -\infty. \end{aligned} \tag{40}$$

Kiekvienam $i = 1, \ldots, N$ ir $j = 1, \ldots, M$ apskaičiuojama kelio radimo kaina

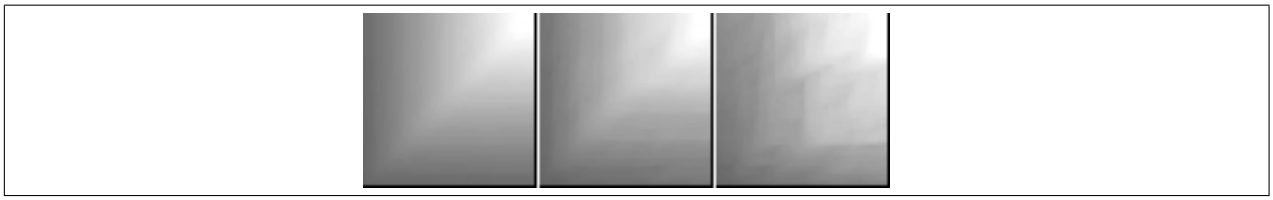
$$cost = \Delta_{dtw}(f_A, f_B), \tag{41}$$

išrenkama mažiausia reikšmė p_{dtw} iš aibės $[M_{dtw}(i-1,j), M_{dtw}(i,j-1), M_{dtw}(i-1,j-1)]$

arba didžiausia, jei maksimizuojamas resultatas. Susumuojami *cost* ir p_{dtw}

$$M_{dtw}(i,j) = cost + p_{dtw}. \tag{42}$$

Ciklišk ai kartojami šie veiksmai, kol $i < N$ ir $j < M$. Rezultatas t. y. galutinis panašumo įvertis, yra $M_{dtw}(i-1, j-1)$.



16 pav.: DTW algoritmo geriausio kelio paieškos matricos. Kairėje yra idealus atitikmuo etalonų įrašui, viduryje - ta pati frazė sakoma to pačio žmogaus kitu laiko momentu, dešinėje - kita frazė, sakoma to pačio žmogaus.

10.3. Rezultatų normavimas

10.3.1. 1 – variantas

Tarkime, turime etalonų duomenų bazę T_{DB}, kurioje yra N įrašų, kurie turi M skirtingų frazių, t. y. turi skirtingus ID.

Apskaičiuojami panašumo įverčiai s įrašo su visa etalonų duomenų baze.

$$\begin{aligned} s_1 &= similarity(A, T_{DB}(1)), \\ s_2 &= similarity(A, T_{DB}(2)), \\ &\dots, \\ s_N &= similarity(A, T_{DB}(N)). \end{aligned} \tag{43}$$

Surandama didžiausia panašumo vertė:

$$s^1_{max} = \max([s_1, \dots, s_N]). \tag{44}$$

Surandamas antras pagal didumą panašumo įvertis iš likusių, kurių ID nesutampa, su kuriuo gavosi s^1_{max}.

$$s^2_{max} = \max(([s^1_{max}] \notin [s_1, \dots, s_N])). \tag{45}$$

Apskaičiuojama rezultatų pernormavimo konstanta:

$$s_{norm} = (s^1_{max} + s^2_{max})/2. \tag{46}$$

Perskaičiuojami galutiniai panašumo įverčiai:

$$\tilde{s_i} = s_i - s_{norm} + \sigma, \tag{47}$$

σ - poslinkis, pavyzdžiui $\sigma = 1000$, naudojamas, kad nesigautų nuliniai rezultatai. Poslinkis turi būti visiems vienodas.

Tokiu būdu normavimas gaunasi individualus kiekvienam testiniam A.

10.3.2. 2 – variantas

Įprastai naudojant DTW algoritmą, pradinis kadras sutapatinamas su pradiniu, po to antras su antru ir t. t. Tokiu būdu gaunamas normalus panašumo įvertis $similarity(A, B)$. Šio normavimo idėja tokia:

Pirmiausia, apskaičiuojamas $similarity'(A, B)$. Apverčiami visi B kadrai t.y. kadrų numeracija vietoje $0, 1, 2, 3,...N$ tampa $N, N-1,...,1, 0$. Tokį šabloną pažymėję B' gauname, kad $similarity'(A, B) = similarity(A, B')$.

Normuotas panašumas skaičiuojamas:

$$similarity_{norm}(A, B) = similarity(A, B) - similarity'(A, B) + \gamma, \tag{48}$$

γ – poslinkis, kuris priklausomai nuo gaunamų rezultatų parenkamas, kad normuotas įvertis neįgytų neigiamos vertės ir γ poslinkis visiems turi būti vienodas.

11. Tyrimas

Sukurto algoritmo tyrimas atliekamas norint įvertinti algoritmo atpažinimo kokybę. Rezultatai pateikiami lentelėmis ir *ROC* kreivėmis. Pateiktos *ROC* kreivės sudarytos pagal du parametrus: *FRR* ir *FAR*.

FRR (ang. False Rejection Rate) – Tai tikimybė, kad atpažinimo sistema atmes autentiškas poras kaip apsišaukėlius.

FAR (ang. False Acceptance Rate) – Tikimybė, kad atpažinimo sistema klaidingai pripažins apsišaukėlius kaip autentiškas poras.

Kadangi analizuojama algoritmo atpažinimo kokybė, o ją aprašančių parametrų yra nemažai, tai parinkti keli esminiai parametrai, kuriais remiantis galima vertinti. Šie parametrai yra: „EER", „ZeroFAR" ir „Rank 1".

EER (ang. Equal Error Rate) – Šis parametras parodo kiek lyginamų porų priėmimo ir atmetimo klaidų lygios, t. y. $FRR = FAR$. Kuo šis parametras mažesnis, tuo atpažinimo sistema tikslesnė.

ZeroFAR – Šis parametras parodo, kokią dalį atpažinimo sistema atmeta autentiškų porų atmesdama visas apsišaukėlių poras, t. y. *FRR* reikšmė, kai $FAR = 0$.

Rank 1 – Šis parametras parodo, kiek porų daugiausiai surinko panašumo taškų.

Eksperimentams buvo naudojama rusų diktorių duomenų bazė – „ELRA-S0050 Russian speech database (STC)", kur du rusiški posakiai kartojami 90 – ies diktorių ir ta pati frazė kartojama 10 – 20 kartų. Etalonų duomenų bazę sudaro 130 įrašų, o realius duomenis imituojančių įrašų – 1005 ir tai sudaro 130156 lyginamų porų. Atpažinimo objektas: sakomas tekstas ir kalbėtojas, t. y. laikoma atpažintu įrašu, kai kalbėtojas ir jo sakomas tekstas sutampa.

11.1. Eksperimentas: Frazių atpažinimas

Šio testo metu buvo tiriama kaip algoritmo atpažinimo kokybė skiriasi keičiant kadro persidengimo dydį. Kadro dydis buvo 256, 512, 1024 ir 2048, o persidengimas kito atitinkamai nuo kadro dydžio 25%, 50%, 75% ir 100%. Šio bandymo metu buvo mėginama nustatyti, kokia įtaką turi atpažinimui kadro ir poslinkio dydis.

Pirmasis bandymas, nenaudojant rezultatų normavimo. Įrašas laikomas atpažintu, kai sutampa kalbėtojas ir sakoma frazė. Gauti rezultatai pateikti lentelėje.

1 lentelė. Bandymas 1: Tiriama kaip priklauso atpažinimo rezultatai nuo lango ir jo persidengimo parametrų dydžio.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	6,42	81	49,9
2	256	50	7,33	79,7	48,7
3	256	75	7,75	90,3	48,5
4	256	100	8,05	91,6	48,4
5	512	25	5	76,9	50
6	512	50	5,19	78	49,5
7	512	75	5,28	80,1	48,9
8	512	100	6,69	86,8	48,2
9	1024	25	3,21	57,5	49,9
10	1024	50	3,77	63,9	49,8
11	1024	75	5,28	60,5	48,9
12	1024	100	7,81	80,3	46,2
13	2048	25	4,99	59,8	49,6
14	2048	50	8,02	78,4	45,1
15	2048	75	17,8	92,4	38
16	2048	100	35,5	99,9	16,2

Iš lentelės matyti, kad esant kadro dydžiui 1024 ir persidengimui 25% gaunami rezultatai (Nr. 9) geriausi remiantis *EER* ir *ZeroFAR* parametrais, kurie atitinkamai yra 3.21% ir 57.5%. Sekantis geriausias rezultatas yra Nr. 10, kur skiriasi tik persidengimo dydis – 50%. Žiūrint į lentelės duomenis matyti, kad esant bet kokiam kadro dydžiui, gaunami geriausi rezultatai, kai persidengimas lygus 25%.

Šio eksperimento antrajame bandyme netikrinkime kalbėtojo, t. y. atpažintas įrašas yra tada kai sutampa tik frazė.

2 lentelė. Bandymas 2: Tiriama kaip priklauso atpažinimo rezultatai nuo lango ir jo persidengimo parametrų dydžio.

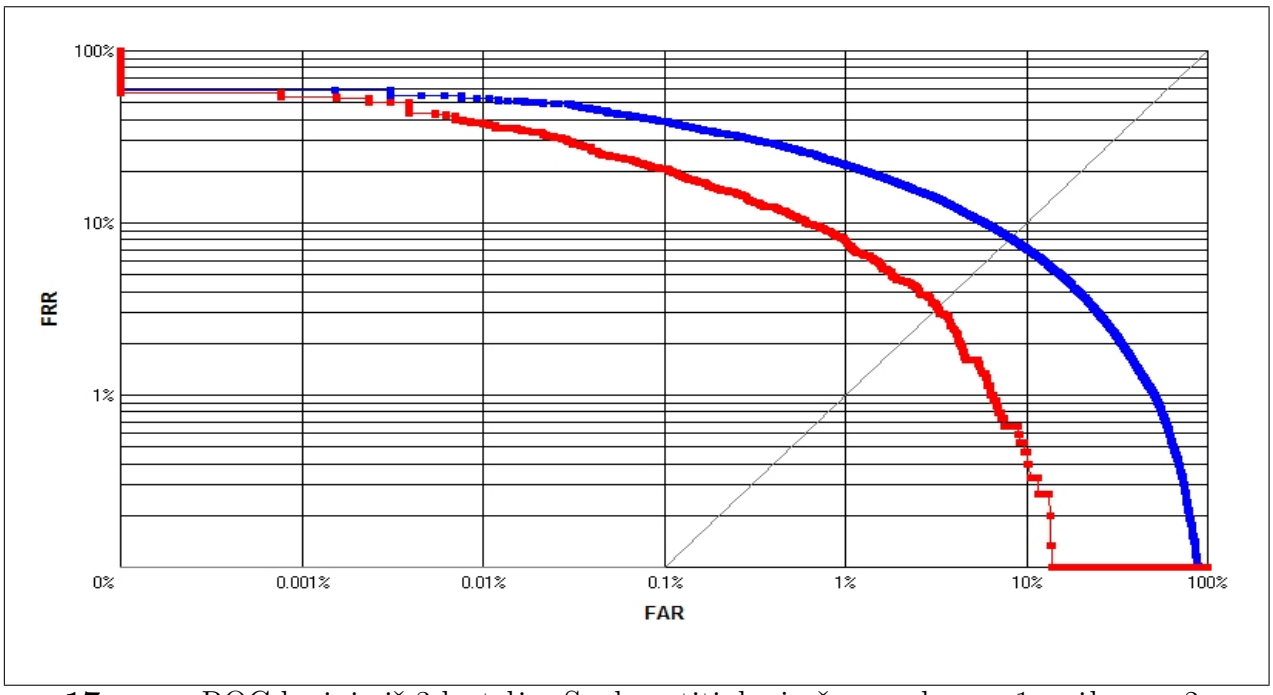
Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	15,1	86,7	50
2	256	50	8,91	68,9	50
3	256	75	9,62	76,9	50
4	256	100	10,4	78,3	50
5	512	25	11,1	75,4	50
6	512	50	10,3	73,8	50
7	512	75	8,59	73,2	50
8	512	100	8,17	58,9	50
9	1024	25	14,4	68,4	50
10	1024	50	13,4	68,2	50
11	1024	75	17,6	92,4	50
12	1024	100	15,8	81,5	50
13	2048	25	21,5	94,5	50
14	2048	50	25,7	92,4	50
15	2048	75	37	99,8	50
16	2048	100	48,2	100	50

Kuomet lyginamos tik frazės, atmetant kalbėtoją, geriausio atitikmens paieška pasunkėja, nes atsiranda daugiau įrašų, kurie gali klaidingai atitikti ieškomą frazę. Iš gautų rezultatų matyti, lyginant su pirmuoju bandymu, kad rezultatai suprastėjo žymiai. Palyginimui paimkime po geriausią rezultatą iš 1 ir 2 lentelės:

3 lentelė. Geriausi rezultatai iš 1 ir 2 lentelių.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	1024	25	3,21	57,5	49,9
2	512	100	8,17	58,9	50

Matyti, kad geriausi rezultatai gaunami prie skirtingų kadro ir jo persidengimo parametrų. Pakito iš esmės tik *EER* parametras, kuris įgijo 8.17 reikšmę. *ZeroFAR* ir *Rank 1* pakito labai nežymiai. Atpažinimo kokybė vienu ir kitu atveju galima geriau palyginti naudojant *ROC* kreivę, kuri pateikta apačioje.



17 pav.: ROC kreivės iš 3 lentelės. Spalva atitinka įrašą: raudona – 1, mėlyna – 2.

11.2. Eksperimentas: Normavimo įtaka algoritmo rezultatams.

Šio testo metu buvo tiriama kaip algoritmo atpažinimo kokybė skiriasi keičiant kadro persidengimo dydį ir kartu naudojant normavimą. Kadro dydis buvo 256, 512, 1024 ir 2048, o persidengimas kito atitinkamai nuo kadro dydžio 25%, 50%, 75% ir 100%. Šio bandymo metu buvo mėginama nustatyti, kokia įtaką turi atpažinimui kadro ir poslinkio dydis, taip pat kaip rezultatai skiriasi panaudojus normavimą.

Pirmasis bandymas, nenaudojant rezultatų normavimo. Gauti rezultatai pateikti lentelėje.

4 lentelė. Bandymas 1: Tyrimas, kaip priklauso rezultatai nuo lango ir jo persidengimo parametrų dydžio.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	6,42	81	49,9
2	256	50	7,33	79,7	48,7
3	256	75	7,75	90,3	48,5
4	256	100	8,05	91,6	48,4
5	512	25	5	76,9	50
6	512	50	5,19	78	49,5
7	512	75	5,28	80,1	48,9
8	512	100	6,69	86,8	48,2
9	1024	25	3,21	57,5	49,9
10	1024	50	3,77	63,9	49,8
11	1024	75	5,28	60,5	48,9
12	1024	100	7,81	80,3	46,2
13	2048	25	4,99	59,8	49,6
14	2048	50	8,02	78,4	45,1
15	2048	75	17,8	92,4	38
16	2048	100	35,5	99,9	16,2

Iš lentelės matyti, kad esant kadro dydžiui 1024 ir persidengimui 25% gaunami rezultatai (Nr. 9) geriausi remiantis EER ir ZeroFAR parametrais.

Antras bandymas, kurio metu buvo panaudotas pirmasis normavimo algoritmas. Šio bandymo metu, buvo tikimasi, kad normavimas pagerins pirmame bandyme gautą geriausią rezultatą (Nr. 9).

5 lentelė. Bandymas 2: Tyrimas, kaip priklauso rezultatai nuo lango ir jo persidengimo parametrų dydžio, naudojant rezultato normavimo 1 – jį algoritmą.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	4,15	76,9	49,9
2	256	50	4,17	82,6	48,8
3	256	75	4,46	86,7	48,5
4	256	100	5,04	82,1	48,4
5	512	25	2,45	76,3	50
6	512	50	2,93	68,7	49,5
7	512	75	3,38	75,1	48,9
8	512	100	4,19	88,5	48,2
9	1024	25	1,86	35,9	49,9
10	1024	50	3,31	63,8	49,8
11	1024	75	5,18	66,5	48,9
12	1024	100	7,11	78,9	46,2
13	2048	25	5,15	56,7	49,6
14	2048	50	9,96	80,1	45,1
15	2048	75	16,4	92,6	38
16	2048	100	32,2	99,9	16,2

Pritaikius normavimą rezultatai ženkliai pagerėjo, *ZeroFAR* ir *Rank 1* parametrai išliko geriausi esant lango dydžiui 1024 ir persidengimas 25%, o atpažinimo tikslumo parametras *EER* lygus 1,86%. Palyginus su 4 lentelės geriausiu rezultatu, *EER* pagerėjo nuo 3.21% iki 1,86%. Kas įdomu taip pat, kad šis normavimas *ZeroFAR* lalbai stipriai paveikė. Palyginus su prieš tai atliktu bandymu, *ZeroFAR* pakito nuo 57,5% iki 35.9%.

Trečias bandymas, kurio metu buvo panaudotas antrasis normavimo algoritmas. Šio bandymo metu, buvo tikimasi taip pat, kad normavimas pagerins pirmame bandyme gautą geriausią rezultatą (Nr. 9).

6 lentelė. Bandymas 3: Tyrimas, kaip priklauso rezultatai nuo lango ir jo persidengimo parametrų dydžio, naudojant rezultato normavimo 2 – jį algoritmą.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	5,17	72,7	50
2	256	50	5,77	75,7	48,9
3	256	75	6,24	88,5	48,5
4	256	100	7,29	95,7	48,1
5	512	25	3,7	70,4	50
6	512	50	3,94	70,2	49,8
7	512	75	4,67	78,9	49,1
8	512	100	6,26	79,8	48,3
9	1024	25	2,45	51,8	49,9
10	1024	50	3,18	54,4	49,8
11	1024	75	4,21	61,8	49,3
12	1024	100	6,91	83,2	46,6
13	2048	25	4,01	62,2	49,5
14	2048	50	5,96	75,9	47,4
15	2048	75	10,9	96	42,5
16	2048	100	20,7	99,3	28,9

Šio bandymo metu, kaip ir buvo tikėtasi, geriausias rezultatas iš 4 lentelės Nr. 10 šio bandymo metu pagerėjo panaudojus 2 – jį normavimo algoritmą. Rezultatas EER pagerėjo nuo 3.21 % iki 2.45 %. ZeroFAR ir Rank 1 parametrai taip pat pakito į teigiamą pusę, bet labai nežymiai. Apskritai 1 – sis algoritmas duoda geresnius rezultatas palyginus su 2 – algoritmu. Tačiau, kadangi jie veikia abu skirtingu principu, tai jų rezultatai gali skirtis priklausomai nuo sąlygų.

Ketvirto bandymo metu, buvo mėginama kartu naudoti pirmąjį ir antrąjį normavimo algoritmus. Tokia algoritmų kombinacija davė gana rimta pagerėjimą ir tai stipriai pagerino atpažinimo metrikos rezultatus.

7 lentelė. Bandymas 4: Tyrimas, kaip priklauso rezultatai nuo lango ir jo persidengimo parametrų dydžio, naudojant rezultato normavimo 1 – jį ir 2 – jį algoritmus kartu.

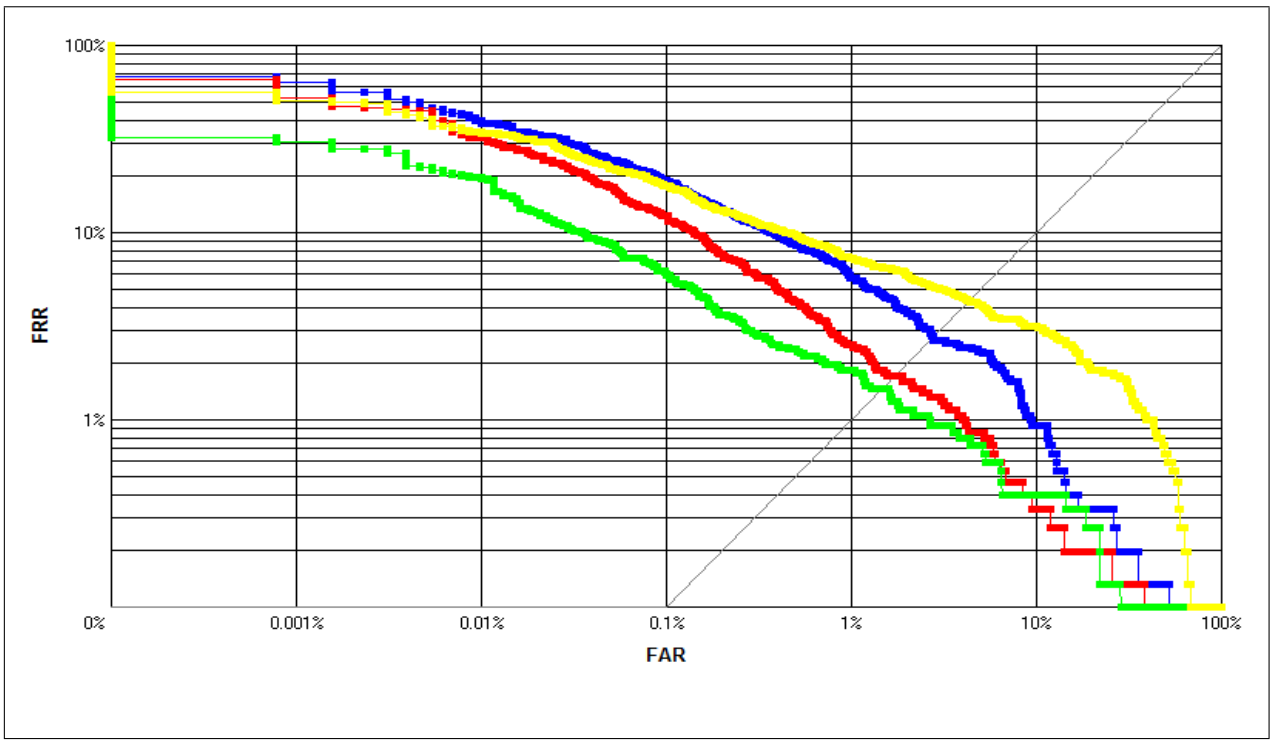
Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	256	25	2,75	67,9	50
2	256	50	3,45	76,5	48,9
3	256	75	3,53	85,1	48,5
4	256	100	4,67	85	48,1
5	512	25	1,72	65,4	50
6	512	50	2,39	54	49,8
7	512	75	2,98	71,5	49,1
8	512	100	4,24	76,9	48,3
9	1024	25	1,46	32,3	49,9
10	1024	50	2,39	49,6	49,8
11	1024	75	3,85	70,4	49,3
12	1024	100	5,7	87,3	46,6
13	2048	25	4,31	56,3	49,5
14	2048	50	6,76	70,2	47,4
15	2048	75	13,2	96,4	42,5
16	2048	100	27	99,3	28,9

11.2.1. Apibendrinimas

Iš bandymo rezultatų matyti, kad geriausi rezultatai gaunami turint 1024 imčių kadro dydį ir persidengimą 25 %. Kombinuotas normavimas pagerino visus rezultatus, kai kur net labai supanašėja jie. Reikėtu pastebėti, kad gaunami rezultatai yra atvirkščiai proporcingi lango persidengimo dydžiui. Kuo lango persidengimas didesnis, tuo atpažinimo kokybė prastesnė. Šio reiškinio priežastis gali būti tokia, kad per ne lyg didelis informacijos kiekis vietoj to, kad gerintų atpažinimą, jį gadiną. Mažinant persidengimo dydį, sumažinama perteklinės informacijos kiekis, kuris neigiamai veikia atpažinimo procesą.

8 lentelė. Lentelėje pateikta iš kiekvieno bandymo (1, 2, 3, 4) po vieną geriausia rezultatą.

Nr.	**Kadro dydis**	**Persidengimas (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	1024	25	3,21	57,5	49,9
2	1024	25	1,86	35,9	49,9
3	1024	25	2,45	51,8	49,9
4	1024	25	1,46	32,3	49,9



18 pav.: ROC kreivės iš 8 lentelės. Spalva atitinka įrašą: geltona – 1, raudona – 2, mėlyna – 3, žalia – 4.

Tolimesniuose tyrimuose bus naudojami parametrai, kurie geriausiai pasirodė šiuose bandymuose: kadro dydis – 1024 imčių, persidengimas kadro – 25%.

11.3. Eksperimentas: Triukšmo poveikis atpažinimo rezultatams

Šiuo bandymu mėginama išsiaiškinti triukšmo poveikį atpažinimo sistemai. Triukšmo simuliacija buvo atliekama taip: Tarkime, turime garso įrašą U, o N – įrašo U masyvo ilgis.

1. Tuomet apskaičiuojamas signalo vidurkis:

$$u_{avg} = \sum_{i=1}^{N} U_i/N. \tag{49}$$

2. Po to apskaičiuojame įrašo vidutinę energiją:

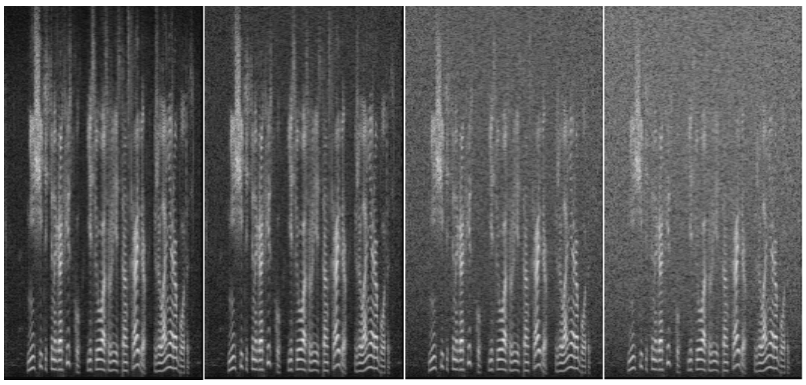
$$e = \sum_{i=1}^{N} |U_i - u_{avg}|/N. \tag{50}$$

3. Tuomet pasirenkame triukšmo lygį p (pavyzdžiui $p = 10$):

$$\tilde{U}_i = U_i + (rand - 0.5) \cdot p/100 \cdot e, \tag{51}$$

kur $rand$ – atsitiktinis skaičius iš realiųjų skaičių intervalo 0 - 1.

$\tilde{U}$ – Audio įrašas su pasirinktu triukšmo lygiu.



19 pav.: Pavyzdys, kaip atrodo spektrograma su triukšmu.(Iš kairės: be triukšmo, su 10% triukšmu, su 50% triukšmu, su 100% triukšmu.)

Bandymas atliekamas generuojant triukšmą aukščiau pateiktu algoritmu. Pagrindinis parametras yra triukšmo lygis, kuris yra įrašo vidutinės energijos procentinė dalis. Eksperimento metu buvo naudojami triukšmo lygiai: intervalas nuo 0% iki 200% su žingsniu 10%, ir atskirai dar bandymas naudojant 400% triukšmo lygį.

9 lentelė. Lentelėje pateikta atpažinimo rezultatai esant triukšmui intervale 0 – 100. Naudojamas kadro dydis 1024 imčių, o persidengimas 25%, nenaudojamas normavimas.

Nr.	**Triukšmo lygis (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	0	3,21	57,5	49,9
2	10	3,02	54,8	49,9
3	20	3,18	55,3	49,8
4	30	3,25	54,7	49,8
5	40	3,31	57,8	49,8
6	50	3,31	58,6	49,8
7	60	3,45	60,7	49,7
8	70	3,51	61,4	49,6
9	80	3,69	65,7	49,6
10	90	3,78	68,9	49,5
11	100	3,98	67,7	49,6

Kuomet triukšmo lygis mažas, t. y. 10% ir 20%, matomas gana žymus rezultatų pagerėjimas. Didinant triukšmo lygį, proporcingai auga *EER* ir *ZeroFar*, tačiau tempas nėra didelis. Vidutiniškai *EER* ir *ZeroFar* padidėja $\sim 3\%$, kai triukšmo lygis padidėja 10%.

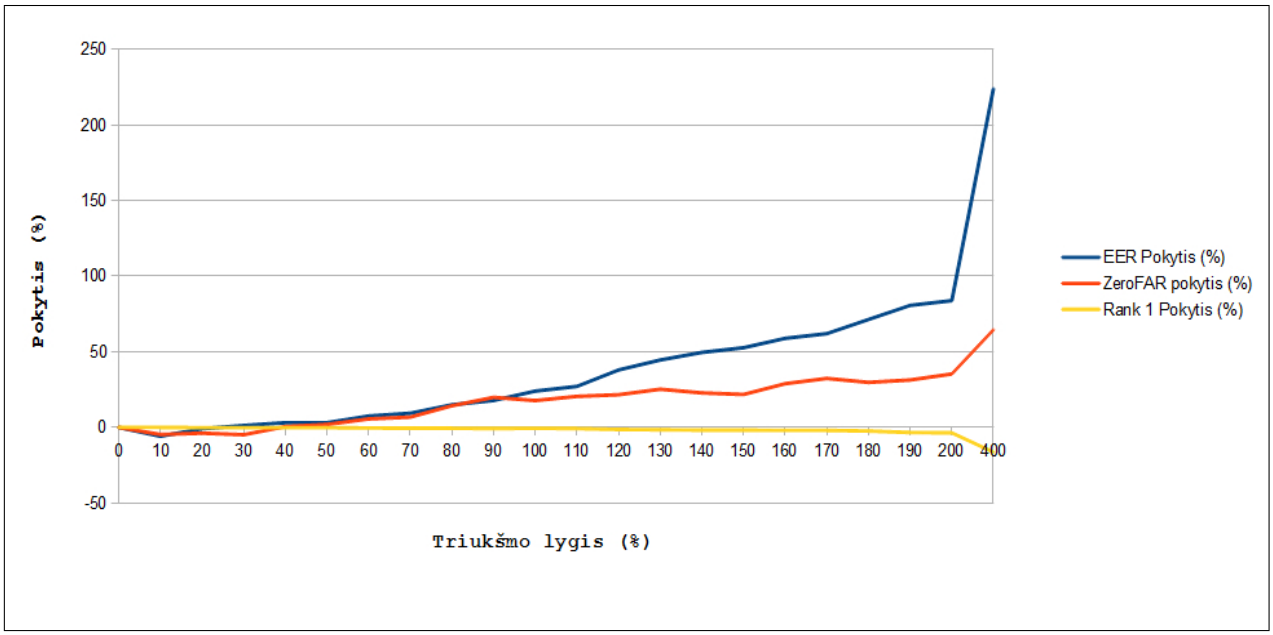
10 lentelė. Lentelėje pateikta atpažinimo rezultatai esant triukšmui intervale 100 – 200. Naudojamas kadro dydis 1024 imčių, o persidengimas 25%, nenaudojamas normavimas.

Nr.	Triukšmo lygis (%)	EER (%)	ZeroFAR (%)	Rank 1 (%)
1	100	3,98	67,7	49,6
2	110	4,08	69,3	49,5
3	120	4,43	69,9	49,2
4	130	4,64	72	49,1
5	140	4,8	70,6	49
6	150	4,9	70	49
7	160	5,1	74,1	48,9
8	170	5,2	76,1	48,9
9	180	5,5	74,6	48,7
10	190	5,8	75,5	48,2
11	200	5,9	77,8	48,1

Triukšmo lygį pakėlus iki 200% matomas tolygus parametrų didėjimas, matomas iš kreivių (Pav. 20). Įdomumo dėlei, buvo atlikta bandymas, kuomet triukšmo lygis lygus 400%.

Nr.	Triukšmo lygis (%)	EER (%)	ZeroFAR (%)	Rank 1 (%)
1	400	10,4	94.6	41,7

Pridėjus šį rezultatą prie jau esamų, matyti, jog labiausiai triukšmo įtakos paveikiamas *EER* parametras, kuris nurodo atpažinimo kokybę.



20 pav.: Kreivės rodančios *EER*, *ZeroFAR*, *Rank 1* kitimą lyginant su pradiniu momentu, kurio metu triukšmo lygis buvo **nulis**.

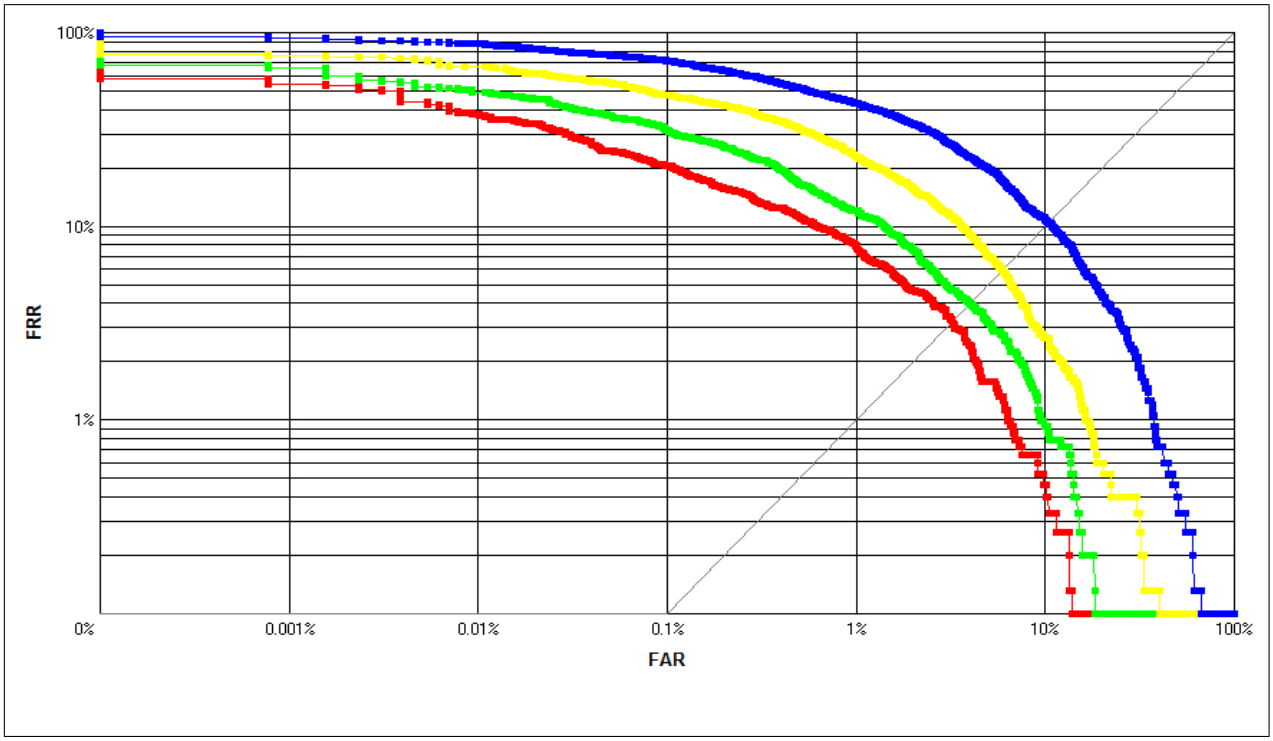
Žiūrint į santikinį rezultatų pokytį išreikštą procentais, matyti kad *EER* kinta greičiausiai, tačiau *EER* nuo pat pradžių buvo mažas ir jo pokytis realiai yra kur kas mažesnis, pavyzdžiui 400% triukšmo lygiui gaunami parametrai labai skirtingi savo reikšmėmis, ypatingai išsiskiria šioje vietoje *ZeroFAR*.

11.3.1. Apibendrinimas

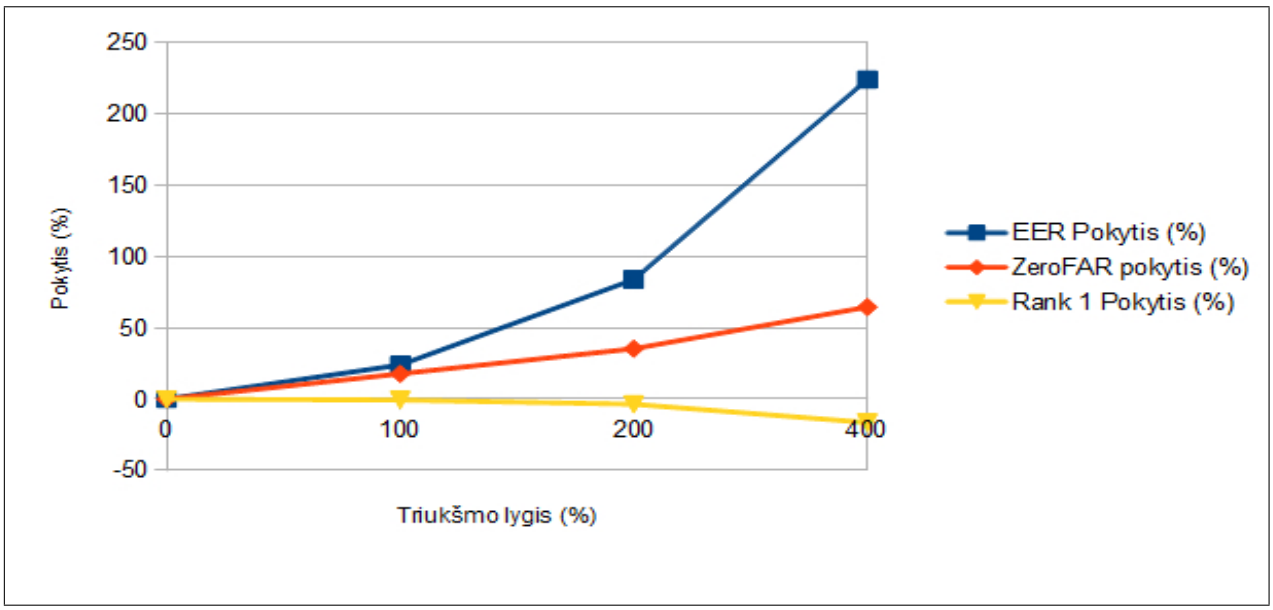
Duomenis pateikiant ne taip smulkiai, lengviau pastebėti, kaip ir kokie parametrai kinta atsiradus triukšmui audio signale. Intervale, kuomet triukšmo lygis yra nuo 0% iki 100% poveikis yra minimalus atpažinimui, tačiau keliant triukšmą, rezultatai pradeda didėti didesniu tempu taip gadinami rezultatus. Tai galima aiškiai pamatyti apačioje pateiktuose duomenyse: tyrimo rezultatai (11 lentelė), ROC kreivės (pav.: 21), parametrų kitimo kreivės (pav.: 22).

11 lentelė. Lentelėje pateikta atpažinimo rezultatai esant triukšmui. Triukšmo lygio kitimas kas 100%. Naudojamas kadro dydis 1024 imčių, o persidengimas 25%, nenaudojamas normavimas.

Nr.	**Triukšmo lygis (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**
1	0	3,21	57,5	49,9
2	100	3,98	67,7	49,6
3	200	5,9	77,8	48,1
4	400	10,4	94.6	41,7



21 pav.: ROC kreivės iš 11 lentelės. Spalva atitinka įrašą: geltona – 1, raudona – 2, mėlyna – 3, žalia – 4.



22 pav.: Remiantis 11 lentele, sudarytos kreivės rodančios *EER*, *ZeroFAR*, *Rank 1* kitimą lyginant su pradiniu momentu, kurio metu triukšmo lygis buvo **nulis**.

11.4. Eksperimentas: Spektrogramos vidurkinimo poveikis rezultatams

Vienas iš kalbos atpažinimo proceso sudedamųjų elementų yra spektrogramos vidurkinimas. Prieš tai atliktuose eksperimentuose vidurkinimo parametrai buvo nustatyti empiriniu būdų, tačiau, ar jie parinkti geriausi, čia jau kitas klausimas. Spektrogramos vidurkinimas susideda iš dviejų parametrų: vidurkinimo spindulio ilgio x – ašimi ir vidurkinimo spindulio ilgio y ašimi, atitinkamai juos pavadinkime x_{radius} ir y_{radius}. Šiais parametrais nurodoma, kiek spektrogramos elementų naudoti vidurkinant ją. Šiam tyrimui buvo atliekamas kalbos atpažinimas, vis su skirtingais vidurkinimo parametrais. Vidurkinimo parametrų reikšmių aibės buvo:

1. $x_{radius} \in [0, 2, \ldots, 8, 10]$,
2. $y_{radius} \in [0, 2, \ldots, 28, 30]$.

Kadangi spektrogramos vidurkinimas dažnių ašimi reikalingesnis nei laiko ašyje, tai atitinkamai ir parametrų reikšmių kitimo ribos skirtingos. Atlikus eksperimentą paaiškėjo, jog geriausi rezultatai gaunami $x_{radius} = 2$, o $y_{radius} = [8, 10, 12, 14]$ reikšmėmis.

11.5. Rezultatai

Iš atliktų eksperimentų matyti, jog kalbėtojo ir frazių atpažinimui šis algoritmas tinka kur kas geriau, nei vien tik pavienių frazių atpažinimui, nors galima jį taikyti ir šiam tikslui. Algoritmo atpažinimo kokybę, ir gan ženkliai, galima pagerinti naudojant normavimo algoritmus, žinoma, tas pagerėjimas gaunamas ne už dyką, nes tai papildomi veiksmai, kurie sulėtina algoritmo darbą. Stiprioji šio algoritmo pusė – atsparumas triukšmui. Netgi esant

200% triukšmui gaunamas labai aukštas atpažinimo rezultatas (12 lentelė), ir nenaudojant normavimo algoritmų.

12 lentelė. Lentelėje pateikta atpažinimo rezultatai esant triukšmui intervale 0 – 400. Naudojamas kadro dydis 1024 imčių, o persidengimas 25%, $x_{radius} = 5$ ir $y_{radius} = 13$, nenaudojamas normavimas.

Nr.	**Triukšmo lygis (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**	**Teisingai atpažinta (%)**
1	0	3,21	57,5	49,9	99,6
2	100	4,44	75,1	49,4	98,6
3	200	6,01	80,7	48,3	96,42
4	400	10,8	97,8	41,9	83,69

Pateiktoje 12 lentelėje rezultatai gauti naudojant spektrogramos vidurkinimo parametrus $x_{radius} = 5$ ir $y_{radius} = 13$, kurie buvo parinkti praktiniu būdu. Spektrogramos vidurkinimas daro nemažą įtaką atpažinimo kokybei, tačiau nuo šių parametrų priklauso ir atsparumas triukšmui. Pateiktoje 13 lentelėje rezultatai gauti tokiu pat būdų kaip ir 12 lentelėje, bet tik sumažinti spektrogramos vidurkinimo parametrai: $x_{radius} = 2$ ir $y_{radius} = 8$.

13 lentelė. Lentelėje pateikta atpažinimo rezultatai esant triukšmui intervale 0 – 400. Naudojamas kadro dydis 1024 imčių, o persidengimas 25%, $x_{radius} = 2$ ir $y_{radius} = 8$, nenaudojamas normavimas.

Nr.	**Triukšmo lygis (%)**	**EER (%)**	**ZeroFAR (%)**	**Rank 1 (%)**	**Teisingai atpažinta (%)**
1	0	2,59	52	50	99,8
2	100	3,94	79.9	49,2	98,21
3	200	6,2	86	47,9	95,62
4	400	10,4	96,2	40,1	80,11

Matyti, jog didėjant triukšmo lygiui, mažėja ir atsparumas. Šiuo atveju sumažėjimas nėra labai didelis, tačiau parinkus x_{radius} ir y_{radius} su žymiai didesniu skirtumu, rezultatų pokytis turėtų turėti tendenciją labiau skirtis esant x_{radius} ir y_{radius} labai mažiems ir esant labai dideliems.

Šaltinio [Kam05] autorius yra pasiūlęs savo kalbėtojo atpažinimo pagal balsą sistemą. Jis pateikia savo pasiūlyto algoritmo rezultatus, kai vieni požymiai (autoriaus pasiūlyti), kurie susideda iš: 4 formančių, 3 antiformančių, signalo pagrindinio dažnio, o kiti – MFCC (ang. Mel-frequency cepstral coefficients).

Count of GMM components	F0	4F	4F3A	4FF0	4F3AF0	MFCC
1	26.65%	-	-	-	-	-
3	24.62%	-	-	-	-	-
5	24.71%	13.8%	10.35%	9.22%	7.44%	7.35%
10	24.6%	12.59%	8.33%	8.25%	5.94%	6.27%
15	-	12.26%	7.99%	8.2%	**5.45%**	**6.15%**
20	-	12.4%	7.62%	8.17%	**5.17%**	**5.86%**
Adaptive	-	12.13%	8.01%	8.13%	**5.66%**	**5.89%**

23 pav.: Šaltinio [Kam05] autoriaus atliktų eksperimentų apibendrinimas.

Atliktų eksperimentų metu gautas gautas geriausias rezultatas panaudojus autoriaus algoritmą ir jo pasiūlytą požymių variantą, gautas $EER = 5.17\%$, o naudojant MFCC – 5.86%. Palyginimui, šiame darbe atliktų tyrimų metu, gautas geriausias $EER = 3.21\%$, o jei naudojamas normavimas, tuomet gaunamas $ERR = 1.46\%$.

Apibendrinant šio darbo rezultatus, galima pasakyti, kad įgyvendintas algoritmas labai gerai atpažįsta kalbėtoją ir jo sakomą frazę. Mėginimas atpažinti vien tik frazę, nekreipiant dėmesio į kalbėtoją, duoda taip pat gana gerus rezultatus, tačiau kaip matyti iš gautų rezultatų, šis algoritmas turi didelį potencialą atpažinti įrašus, kurie susieti su kalbėtoju, t. y. atpažinti kalbėtoją, todėl dominavo ir tyrimai šia kryptimi.

Išvados

Iš eksperimentų rezultatų aiškiai matomas algoritmo efektyvumas, ypač atpažįstant kalbėtoją ir jo sakomą tekstą. Gaunamas $EER = 3.21\%$, panaudojus normavimą gaunamas geriausias rezultatas $EER = 1.46\%$, be to, atpažįstamų teisingai porų yra 99,6%. Šis algoritmas pasižymi atsparumu triukšmui, net esant 200% triukšmo lygiui, gaunamas $EER = 6.2\%$, o teisingai atpažintų porų 96.42%. Sumažinus triukšmo lygį iki 100% EER lygus 4.44% ir teisingai atpažintos 98.6% poros. O tik frazių atpažinimo metu gautas $EER = 8.17\%$. Taigi algoritmo stiprioji savybė – konkrečios frazės, pasakytos konkretaus kalbėtojo, atpažinimas.

Šį darbą papildant, reikėtų išsamiau atlikti vien tik frazių atpažinimo eksperimentų ir tyrimų, gal pavyktų pagerinti atpažinimo rezultatą, taip pat reikėtų pakartoti tyrimą su kita ir didesne duomenų baze. Šiuose tyrimuose buvo naudojami tik vyrų balsai, todėl dar naudinga būtų patikrinti ir atpažinimo kokybę naudojant moterų balsus arba mišrias duomenų bazes.

Apibendrinant, pateiktas algoritmas pasižymi aukštu atpažinimo tikslumu ir dideliu atsparumu triukšmui. Reikėtų paminėti, kad šis algoritmas turi greitaveikos problemų, tačiau naudojant šiuolaikines programavimo technikas, pavyzdžiui darbo išskaidymas į mažesnius atskirus darbus, kurie būtų atliekami atskirose gijose, galimas žymus algoritmo darbo pagreitėjimas.

Literatūros sąrašas

[Bas04] dr., doc. A. Bastys, „Fonemų atpažinimas“, Vilniaus Universitetas, 2004, URL: *https://kedras.mif.vu.lt/bastys/academic/ATE/LPC/FonAtp.htm*

[Bac04] T. Bäckström, „Linear predictive modelling of speech – constraints and line spectrum pair decomposition “, 2004, p.25 – p.32, p.33 – p.36, p.40

[BBL98] J. Benharouga, R. von Borries, D. Ljunggren, P. Loughmiller, M. Wagstaff, „Methods for Speech Recognition“, A Project for the Spectral Analysis course ELEC 532. Rice University, 1998, URL: *http://www.clear.rice.edu/elec532/PROJECTS98/speech/*

[CF00] S. De Curtis, J. F. Ferrer, „Fizika“, Šviesa, 2000, p.28.

[JAB09] Douglas L. Jones, S. Appadwedula, M. Berry, M. Haun, J. Janovetz, M. Kramer, D. Moussa, D. Sachs, B. Wade, „Speech Processing: Theory of LPC Analysis and Synthesis“, 2009, URL: *http://cnx.org/content/m10482/2.19/*

[JR91] B. H. Juang, L. R. Rabiner, „Hidden Markov Model for Speech Recognition“, *Technometrics*, 1991, p.251 – p.272.

[Kal04] D. Kallulli, „Linguistics in Automatic Speech Recognition: Problems and Challenges“ skaidrės, University of Vienna, 2004, URL: *http://homepage.univie.ac.at/dalina.kallulli/downloads/Ottawa.pdf*

[Kam05] J. Kamarauskas, „Speaker recognition by voice“, Vilnius Gediminas Technical University Institute of Mathematics and Informatics, 2009.

[Kul05] A. Kulionis, „Kalbos signalų analizės priemonių tyrimas“, Vilniaus Pedagoginis Universitetas, 2005.

[MicRea] „Acoustic Modeling“, URL: *http://research.microsoft.com/en-us/projects/acoustic-modeling/default.aspx*

[MGR05] J.E. Munoz-Exposito, S. Garcia-Galan, N. Ruiz-Reyes, P. Vera-Candeas and F. Rivas-Pena, „Speech/music discrimination using a single warped LPC–based feature“, 2005, Electronics and Telecommunication Engineering Department, University of Jaen, URL: *http://ismir2005.ismir.net/proceedings/2027.pdf*

[Moo05] Andrew W. Moore, Carnegie Mellon University, „Hidden Markov Models“, School of Computer Science, 2005, URL: *http://www.autonlab.org/tutorials/hmm14.pdf*

[Pik06] J. Pike, „Automatic Speech Recognition Techniques“, 2006, URL: *http://www.globalsecurity.org/intell/systems/asr-tech.htm*

[Rab89] L. R. Rabiner, „A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition“, 1989, p.257 – p.261.

[Urb11] A. Urbanavičius, „Fonetikos sąvoka“, 2011, URL: *http://ualgiman.dtiltas.lt/fonetikos_savoka.html*

[wiki11a] Wikipedia, „Speech recognition“, 2011, URL: *http://en.wikipedia.org/wiki/Speech_recognition*

[wiki11b] Wikipedia, „Garsas“, 2011, URL: *http://lt.wikipedia.org/wiki/Garsas*

[wiki11c] Wikipedia, „Akustinis triukšmas“, 2011, URL: *http://lt.wikipedia.org/wiki/Triuk%C5%A1mas#Akustinis_triuk.C5.A1mas*

[wiki11d] Wikipedia, „Pulse Code Modulation“, 2011, URL: *http://en.wikipedia.org/wiki/Pulse-code_modulation*

[Zha10] Le Zhang, „Statistical Language Modeling“, University of Edinburgh, 2010, URL: *http://homepages.inf.ed.ac.uk/lzhang10/slm.html*