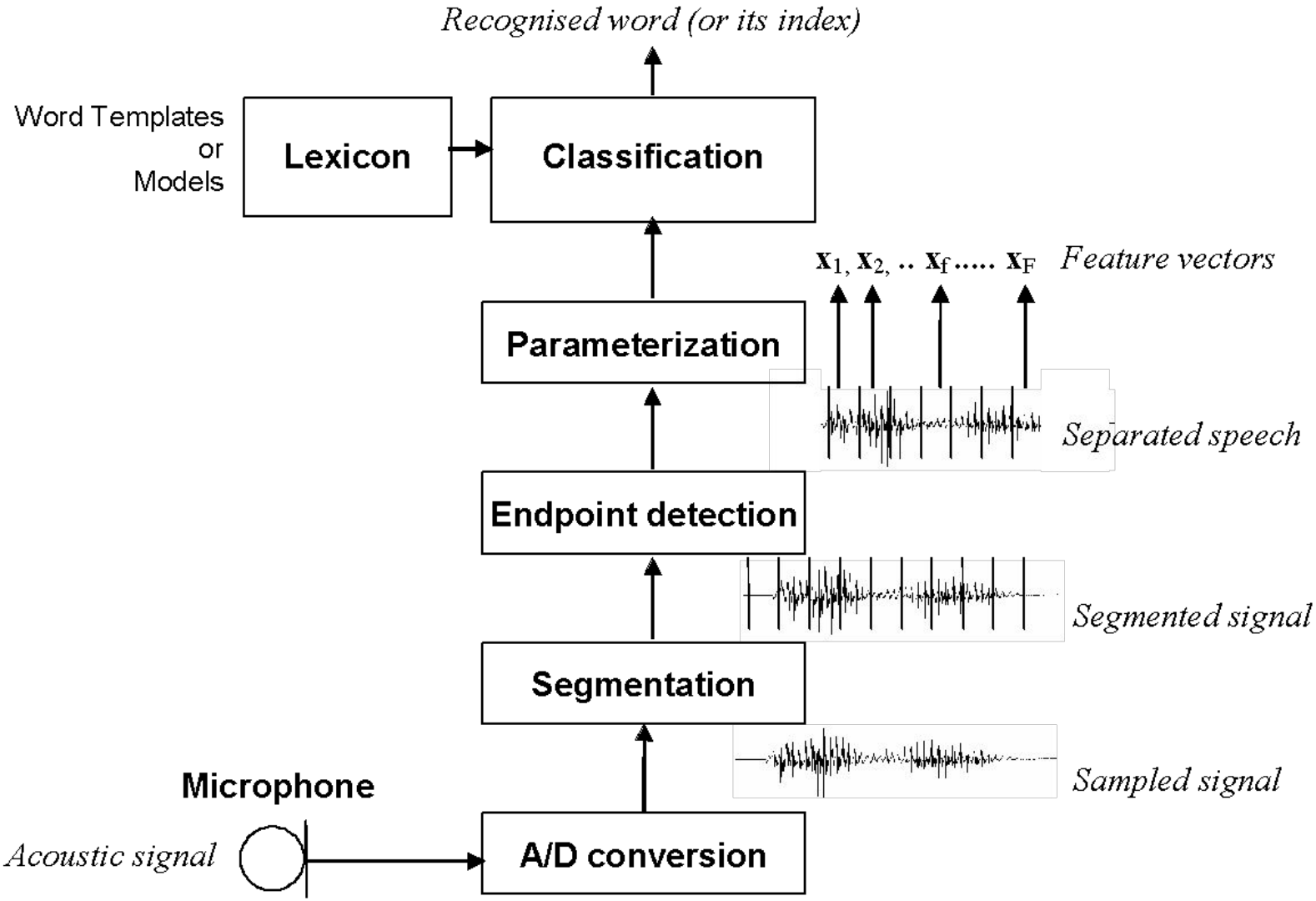
Počítačové zpracování řeči

Přednáška 3

Rozpoznávání slov metodou LTW

Připomenutí - základní kroky IWR



Připomenutí – již zvládnuté kroky

V rámci předchozí úlohy jste zvládli:

- nahrání signálu a jeho segmentaci na framy dlouhé 400 vzorků (framy mají začátky posunuté o 10 ms tj. o 160 vzorků), 2s dlouhá nahrávka má tedy 197 plnohodnotných framů (u posledních 3 už by scházely vzorky)
- výpočet energie a ZCR v každém framu
- nalezení zač. a konce řeči v každé nahrávce na základě průběhu energie
- přípravu dat pro testování i trénování (tvorbu referencí)

Nyní máte pro každou nahrávku k dispozici:

- sekvenci **T** framů, které už obsahují jenom řeč
- v každém z těchto framů jste určili hodnoty 2 příznaků: Ene a ZCR

Když se podíváme na hodnoty **T** (délka slova v nahrávce v počtu framů), vidíme, že **T** je různé nejen pro různá slova ale i pro stejná slova řečená jak stejnými tak i různými osobami

Měření podobnosti

Každý frame řečového signálu je reprezentován:

vektorem **x** = [příznak$_1$, příznak$_2$, .. příznak$_P$] např. **x** = [Energy, ZCR]

Celé slovo je pak reprezentováno:

maticí **X** = [$\mathbf{x_1}$, $\mathbf{x_2}$, ...$\mathbf{x_T}$]

Matice má rozměr *P x T*, *P* je vždy konstantní (počet příznaků)
T se mění (v závislosti na délce slova)

příznaky

x_{11}	x_{21}	x_{31}	..	..	x_{T1}
x_{12}	x_{22}	x_{32}			x_{T2}
..	..	..			..
x_{1P}	x_{2P}	x_{3P}	..	..	x_{TP}

framy →

Měření podobnosti mezi reprezentacemi neznámého slova X a referencemi Ri (tj. reprezentacemi slov získanými ve fázi trénování) se převádí na **výpočet vzdálenosti** mezi maticemi **X** a **Ri**

Měření vzdálenosti

Problém různých délek:

slovo **X** má délku **I** (framů) reference **R** má délku **J** (framů)

Uvažujme nejprve speciální případ kdy I = J

pak **vzdálenost D** mezi slovem **X** a referencí **R** se definuje jako

$$D(\mathbf{X},\mathbf{R}) = \sum_{i=1}^{I} d(x_i, r_i)$$

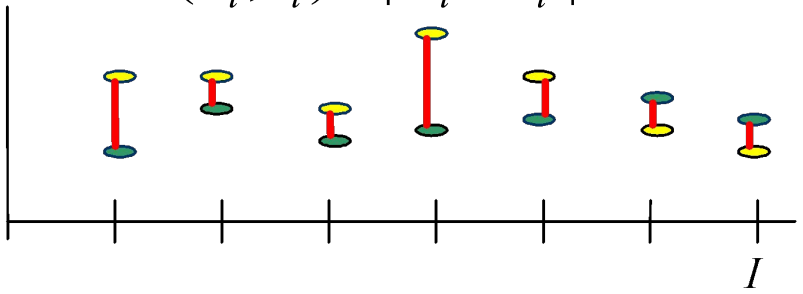
i …. index framu

kde $d(x_i, r_i)$ je ***lokální*** vzdálenost mezi dvěma vektory o dimenzi *P*

nejčastěji se používá **Euklidovská vzdálenost** $d(x_i, r_i) = \sqrt{\sum_{p=1}^{P} (x_{ip} - r_{ip})^2}$

ve speciálním případě, kdy P = 1: $d(x_i, r_i) = | x_i - r_i |$

Ilustrace:
hodnoty x, hodnoty r
lokální vzdálenost



Lineární časová transformace (1)

Uvažujme nyní obecný případ kdy I ≠ J

Řešení spočívá v následující strategii:

Prodluž kratší, či **zkrať delší** reprezentaci tak, aby obě měly **shodnou délku**. Pak už lze použít metodu měření vzdáleností představenou na předchozí stránce. Této metodě, která modifikuje délku, se říká ***časová transformace***, nebo častěji **borcení časové osy (time warping).**

Lineární časová transformace (LTW):

Nejčastější způsob spočívá v **opakování či vynechání** některých framů.

(Opakování *prodlouží*, vynechání *zkrátí* reprezentaci, počáteční a koncové framy si musí odpovídat.)

Výběr framů, které budou takto modifikovány, určí **lineární vztah**:

$$j = w(i) = Int[\frac{J-1}{I-1} \cdot (i-1) + 1 + 0.5] \qquad i = 1, 2, \ldots I$$

Jedná se o rovnici „přímky“ procházející body (1,1) a (I,J)

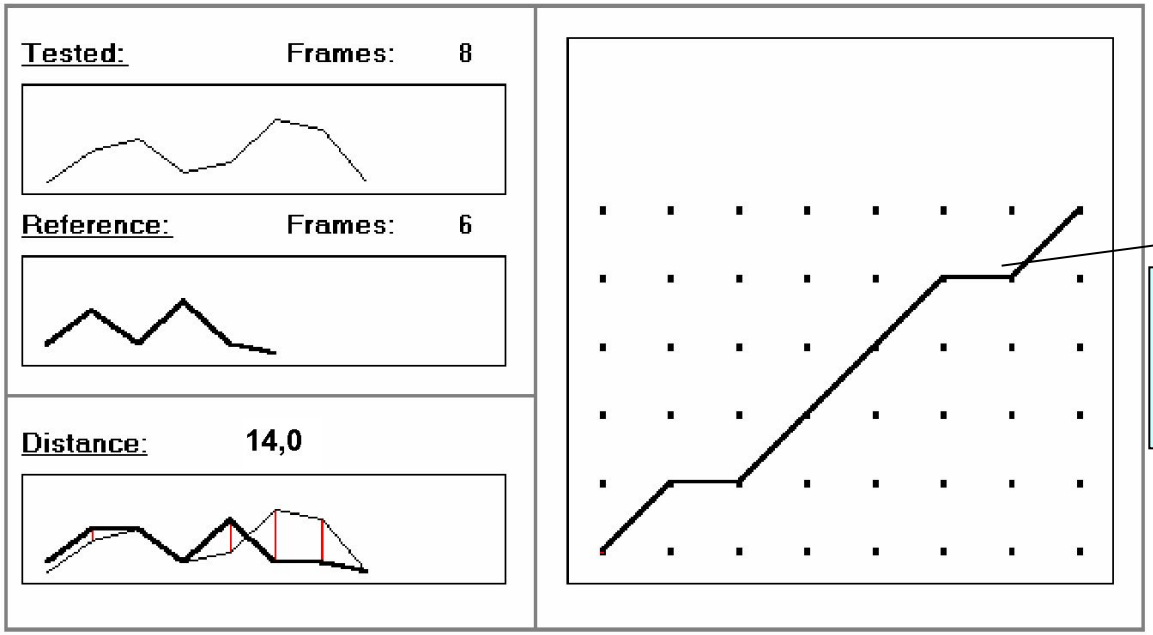
Pak lze určit vzdálenost dvou slov: $$D(\mathbf{X}, \mathbf{R}) = \sum_{i=1}^{I} d(x_i, r_{w(i)})$$

Lineární časová transformace (2)

Příklady a ilustrace:

Uvažujme P = 1 a reprezentace **x = (1, 4, 5, 2, 3, 7, 6, 1)** I = 8

r = (2, 5, 2, 6, 2, 1) J = 6



Globální vzdálenost

Lokální vzdálenost

Transformační funkce („cesta“) - w(i)

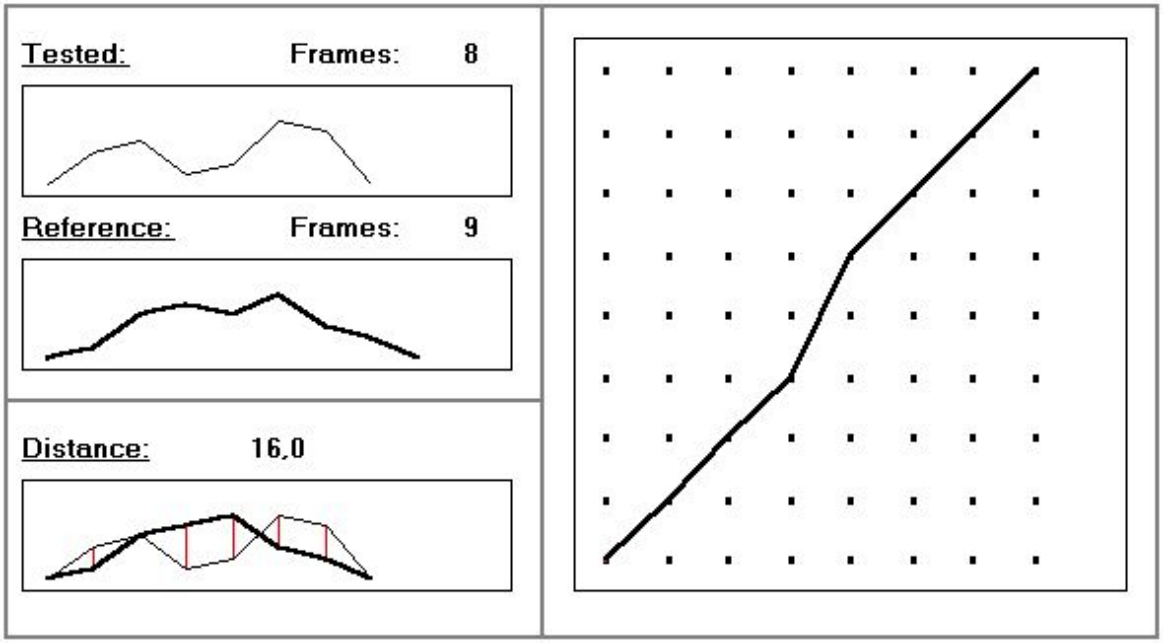
Zopakovaný frame reference

Lineární časová transformace (3)

Jiný příklad:

x = (1, 4, 5, 2, 3, 7, 6, 1) I = 8

r = (1, 2, 5, 6, 5, 7, 4, 3, 1) J = 9



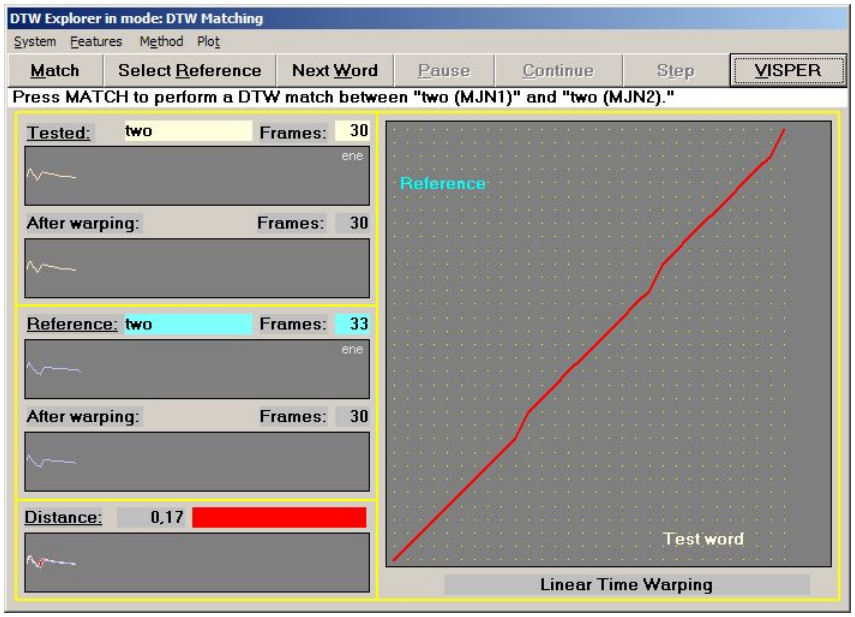
Vynechaný frame reference

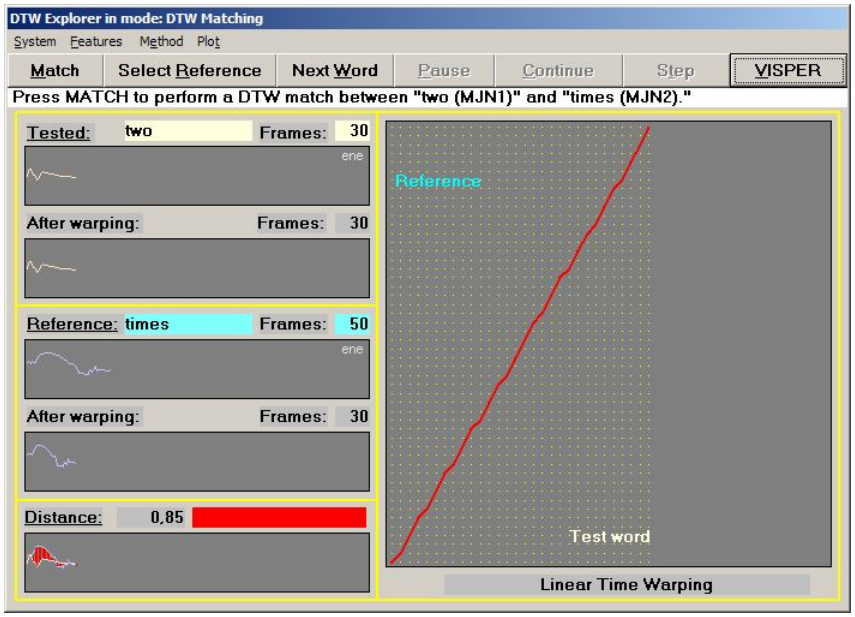
LTW v programu VISPER (1)

Příklady reálných slov – ilustrace v programu VISPER

Slovo “two” (reprezentované jediným příznakem - energií) porovnáváno s

referencí “two” referencí “times”



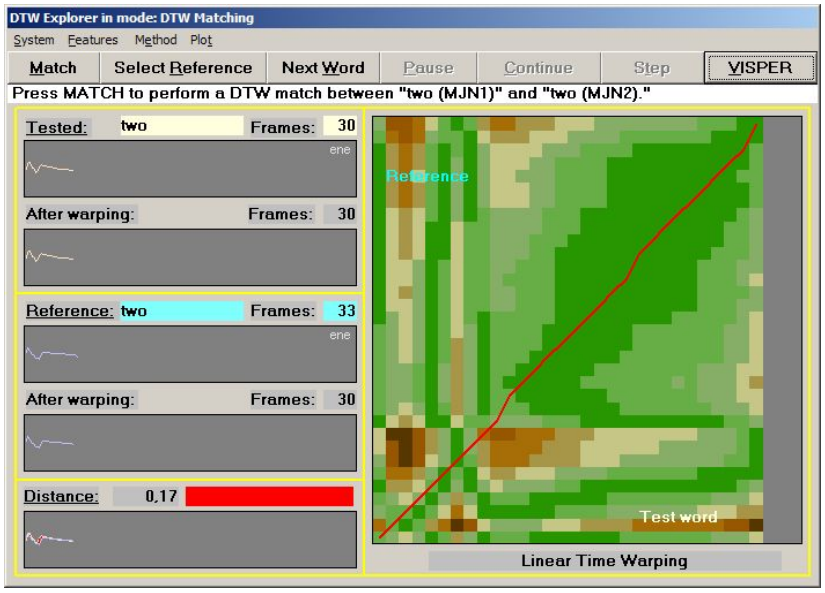


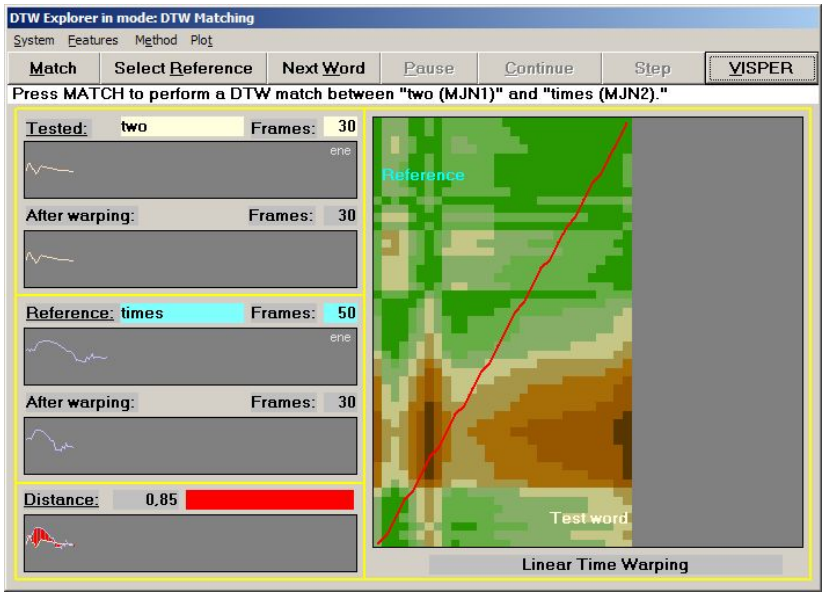
LTW v programu VISPER (2)

Příklady reálných slov– zobrazení pomocí “barevné mapy”

Slovo “two” (reprezentované jediným příznakem - energií) porovnáváno s

referencí “two” referencí “times”





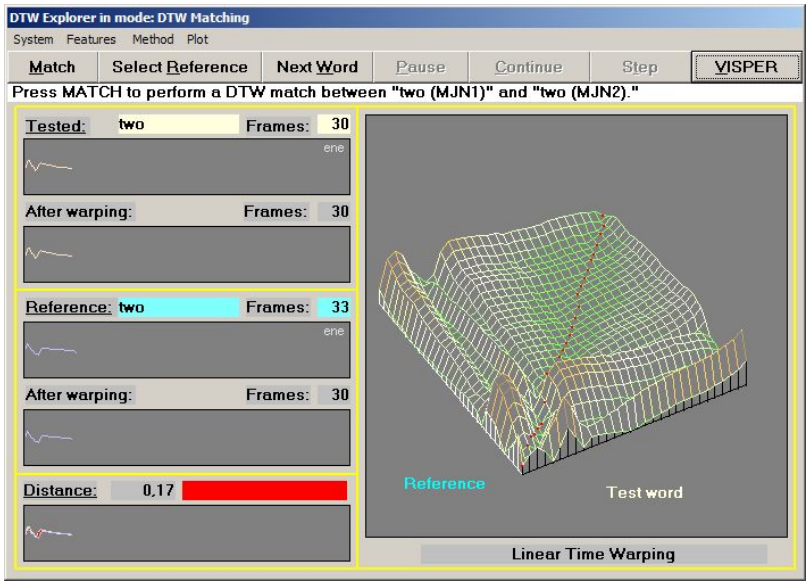
Lokální vzdálenost mezi dvěma framy zobrazena barvou –
zelená = malá vzdálenost, hnědá = velká vzdálenost

LTW v programu VISPER (3)

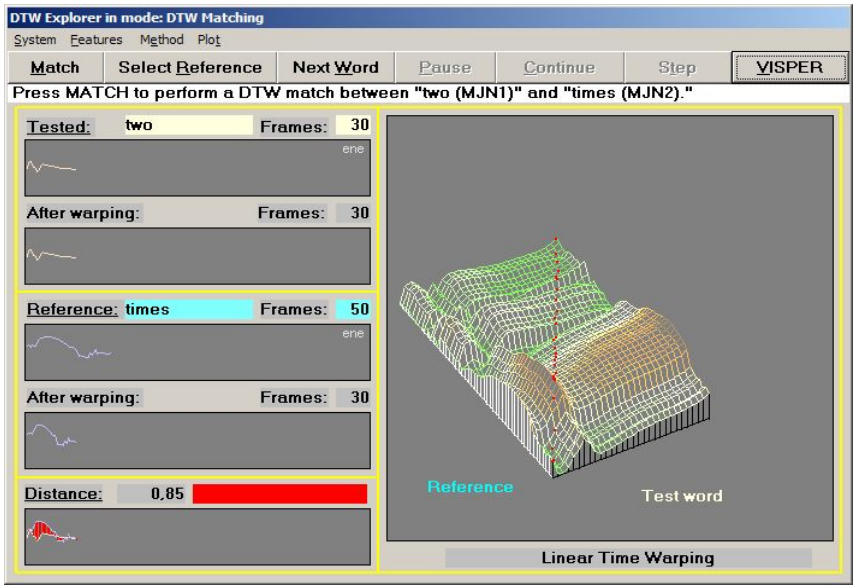
Příklady reálných slov – zobrazení pomocí “3D terénu”

Slovo “two” (reprezentované jediným příznakem - energií) porovnáváno s

referencí “two”



referencí “times”



Lokální vzdálenost mezi dvěma framy zobrazena barvou a výškou terénu

Úloha pro cvičení

Rozpoznávač číslic založený na LTW

1) Využijte nahrávky číslic z minula.
2) U každé nahrávky proveďte předzpracování, rozdělení na framy a najděte framy se začátkem a koncem řeči (pomocí log. energie)
3) Pro řečové framy spočítejte následující příznaky:
   log. energie, ZCR, K spektrálních příznaků (postup popsán dále)
4) Takto zparametrizovaná data rozdělte na trénovací a testovací
5) Implementujte algoritmus LTW, proveďte rozpoznávací testy a vyhodnoťte pro různé sady příznaků (E, E+ZCR, Spektrum) a shrňte v tabulce výsledků

Tipy pro implementaci (1)

Parametrizace (výpočet příznaků)

1. Parametrizaci provádějte u každé nahrávky už jen pro framy odpovídající řeči
2. Proveďte 3 typy parametrizace:
   - 1 příznak - pouze Ene
   - 2 příznaky – Ene + ZCR
   - K příznaků – K spektrálních hodnot
3. Pro spektrum použijte následující postup (aplikovaný na každý frame)
   - frame tvoří 400 vzorků, doplňte nulami na 512 hodnot
   - vynásobte těchto 512 hodnot Hammingovým oknem o délce 512
   - vypočtěte spektrum pomocí 512-bodové FFT
   - vezměte prvních 256 hodnot a určete jejich amplitudu (abs. hodnotu)
   - spočítejte z nich K spektrálních příznaků tak, že 256 amplitudových hodnot rozdělíte do K pásem a v každém spočítejte střední hodnotu
   - zvolte K= 16

Tipy pro implementaci (2)

Vytvoření experimentálního prostředí

- Vytvořte si program v Matlabu, který bude pracovat s dodanými daty a vlastními moduly a který vám umožní provádět rozsáhlejší experimenty.
- Program by měl načíst všechna data ze souboru FileList.txt (viz ukázka), vyříznout z nich řeč (detektor), zparametrizovat ji podle potřeby, rozdělit data na trénovací a testovací část, provést požadované experimenty a vypsat výsledky.
- Vhodnými přepínači v programu můžete volit např. použité příznaky, nastavovat parametry řečového detektoru, volit použité metody rozpoznávání (zatím LTW, příště přibydou další), atd.
- Takto připravené prostředí s výhodou využijete i v dalších úlohách.

Tipy pro implementaci (3)

Implementace LTW algoritmu

- **Reprezentace neznámého slova** musí být vždy **na ose i (vodorovné).** Tím se zajistí, že se budou modifikovat pouze reference, a to vždy na **stejnou délku** rozpoznávaného slova. Pak je možné porovnávat vypočtené vzdálenosti a hledat tu, která je nejmenší.

- Implementaci LTW algoritmu si odlaďte na jednoduchých příkladech z této přednášky, pro kontrolu si graficky zobrazte výsledky a cestu

- Je dobré implementovat LTW jako **funkci** s parametry, např. ComputeLTW (X, I, R, J, P)

  kde .. P je počet příznaků
  X je sekvence příznakových vektorů slova s I framy
  R je sekvence příznakových vektorů reference s J framy

- Stejně tak je vhodné implementovat **speciální funkci** pro výpočet **Euklidovské (případně jiné) vzdálenosti** pro daný počet příznaků, př. ComputeEuclidDist (x, r, P)

  kde P je počet příznaků, x a r jsou 2 příznakové vektory

Tipy pro implementaci (4)

Rozpoznávání a testování

- Provádějte testy pro vámi nahranou osobu a dále pro všechny osoby ze sad 2023, 2022 a 2021 z elearningu
- U každé osoby zvolte trénovací set (reference budou slova ze sady 1) a testovací set (sady 2-5)
- Proveďte experimenty se všemi sety příznaků a sestavte tabulku výsledků pro každou osobu a průměr přes všechny osoby
- U příznakové sady Ene + ZCR si všimněte, v jakých rozmezích se tyto 2 příznaky pohybují. Usuďte jaký to má vliv na výpočet vzdálenosti a zkuste vymyslet, co by se s tím dalo dělat, abyste dostali lepší výsledek. Úvahu experimentálně ověřte.

Zaslání programu a výsledků

Rozpoznávání a testování

- V zipu mi pošlete váš program a tabulku výsledků – pro každý typ příznaků výsledky pro každou osobu a průměr
- Lze očekávat úspěšnost od cca 30 % (pouze energie) ale i nad 60 % (zejména spektrální příznaky). Hodnoty budou záviset na konkrétních datech (zejména na kvalitě vašich nahrávek a výslovnosti) a do velké míry i na tom, **jak dobře máte stanoven začátek a konec slov**.
- Experimenty můžete provádět s různými hodnotami prahu (pro stanovení začátku a konce slov) a najít takto jeho nejlepší hodnotu.
- Váš program otestuji tak, že mu předložím vlastní FileList.txt (stejný jako v ukázce na elearningu), a měl by okamžitě fungovat.
- Řešení mi pošlete nejdéle do pondělí 12.00