

Základní zpracování řeči v C

Jan Černocký, FIT VUT Brno

Úkolem tohoto kursu je naučit se implementovat algoritmy nejen v Matlabu (po malém zacvičení to jde samo ;-), ale také v jazyce C, který je průmyslovým standardem. V tomto cvičení se seznámíme se základním čtením zvuku ze souborů, dělením na rámce a výpočtem základních parametrů.

Vstupem zvuku budou soubory RAW (bez hlavičky) nebo WAV (s hlavičkou), výstupem budou binární soubory s jedním, případně vektorem výsledků pro jednotlivé rámce. Tyto výsledky budou zapisovány jako čísla typu float.

Pro kompilaci a hraní s programy doporučujeme Linux, pokud jste zruční v C-Buildru, Visual-C++ a jiných prostředích pod Windows, nebráníme Vám je v laboratořích počítat, ale cvičící pravděpodobně nebudou schopni zodpovědět otázky typu “nejede mi to, protože chybí knihovna ržýářžýá.dll !”

1 Wire

Předtím, než začnete cokoli zpracovávat, je vhodné si napsat software, který jen okopíruje vstup na výstup a jinak nedělá nic. Toto je realizováno v programu wire (drát).

1.1 Instalace

- ze [www-stránky kursu](#) si stáhněte balík `wire.tgz` (je v něm i testovací signál `test.116` “létající prase”).

- rozbalte: `tar xvzf wire.tgz`

- zkompilejte:

```
cd wire
make clean
make
```

- a zkuste pustit, např.: `./wire test.116 /tmp/koko.frames`

- měli byste vidět: `test.116 -> /tmp/koko.frames 132 frames.`

1.2 Chodí to ?

V souboru `/tmp/koko.frames` by teď mělo být uložených 132 rámců každý po 240-ti vzorcích (viz parametr `WIN_LEN` v `config.h`) ve formátu float. Bude dobré načíst v **Matlabu** výsledek a podívat se na něj:

```
% cteni vysledku - koukneme na 13. ramec.
ff = fopen('/tmp/koko','r');
m = fread (ff, [240 inf], 'float');
fclose (ff);
subplot (211); plot (m(:,13));
% vyroba tehoz v Matlabu
ff = fopen('test.116','r');
s = fread (ff, [1 inf], 'short');
mm = frame (s,240,160) / 32768.0; % musime normovat, wire to taky dela...
subplot (212); plot (mm(:,13));
```

Oba dva rámce by měly být stejné, můžete to zkusit i exaktně pomocí `mm(:,13) - m(:,13)`

1.3 Kuk do wire

Podívejte se wiru na zoubek. Hlavním souborem je podle předpokladu `wire.c`. Po otevření souboru se zvukem se spustí hlavní cyklus, který probíhá, dokud se funkci `new_frame` daří naplnit celý rámec vzorky. Podívejte se na funkci `new_frame` do `wavframes.c` a uvědomte si, že její chování musí být trochu jiné pro nultý a ostatní rámce:

- pro nultý rámec musí načíst `WIN_LEN` vzorků.
- pro další rámce musí staré vzorky posunout doleva a načíst pouze `WIN_SHIFT nových` vzorků.

Vlastní zpracování se děje ve funkci `process_frame` (soubor `speech.c`) – v této verzi je to obyčejné kopírování vzorků z pole `frame` do pole `params`.

Úkol

1. Vyzkoušejte, zda `wire` funguje i s WAV soubory. V Matlabu můžete pro čtení WAVka použít s výhodou funkci `wavread`. Tato funkce se sama postará o normování do intervalu ± 1 .

2 Výpočet parametrů

Po ověření “drátem” se dáme do něčeho inteligentnějšího — výpočtu energie a výpočtu autokorelačních koeficientů.

2.1 Úkol 1 – Energie

Modifikujte software pro výpočet krátkodobé energie pro každý rámec podle vzorce:

$$E = \frac{1}{l_{ram}} \sum_{n=0}^{l_{ram}-1} x^2[n] \quad (1)$$

Mimo jiné bude potřeba změnit `LEN_PARAMS` a to na 1 (máme pouze jediný parametr na rámec).

Srovnání v Matlabu pak bude vypadat asi takto (předpokládáme, že signál `s` a rámce `mm` máte v Matlabu pořád k dispozici):

```
% cteni vysledku - energie
ff = fopen('/tmp/energie','r');
e = fread(ff, [1 inf], 'float');
fclose(ff);
subplot(211); plot(s); axis tight;
subplot(212); plot(e); axis tight;
% pocitani energie v Matlabu
em = sum(mm.^2) / 240; % podumejte chvili, jak tato radka chodi !
subplot(213); plot(em); axis tight;
```

Měli byste dostat stejné výsledky. Pokud bude tvar sedět, ale numericky budete jinde, zapřemýšlejte, zda jste někde neudělali chybu v normalizaci nebo v dělení.

2.2 Úkol 2 – Autokorelační koeficienty

Odhadněte v každém rámci autokorelační koeficienty $R[0] \dots R[10]$ podle vzorce:

$$R[k] = \sum_{n=0}^{N-1-k} s[n]s[n+k]$$

Srovnání v Matlabu pak bude vypadat asi takto (předpokládáme, že signál s a rámce mm máte v Matlabu pořádku k dispozici):

```
% ctení výsledku - autokokorelace
ff = fopen('/tmp/autokorelace','r');
R = fread(ff, [11 inf], 'float');
% vybereme si pro kontrolu napr. 13. frame:
r = R(:,13);
% autokorelace v Matlabu - 13. ramec
rm = xcorr(mm(:,13));
rm = rm(240:250); % proc tohle delame ???
subplot(211); plot(0:10,r)
subplot(212); plot(0:10,rm)
```

Výsledky by měly být stejné. Autokorelační koeficienty budeme potřebovat v následujícím cvičení pro lineární predikci.

3 FFT aneb “nagooglit, stáhnout, modifikovat, použít”

Tato část cvičení se zabývá velmi typickou úlohou — máte za úkol napsat nebo nagooglit a stáhnout funkci, která něco dělá, a interfacovat ji s Vaším softwarem. Máte za úkol spočítat spektrogram. V každém rámci tedy musíte spočítat modulové spektrum. V souboru `fft.c` naleznete funkci `ModuleFFT`, která to pro Vás udělá, ale:

- Umí počítat jen s délkami vektoru, které jsou mocninou 2. My chceme zachovat `WIN_LEN=240`. Pamatujete si z `SXC/ISS`, co dělat v takovém případě? Doplnujeme na potřebnou délku nulami (zero-padding).
- Na tento doplněný rámec musíte alokovat (vlastně stačí deklarovat...) pomocné pole o délce `FFT_LEN`.

Při kontrole v Matlabu si jednak “pohrejte” s jedním rámcem (doporučuji opět 13., kde je pěkně vidět harmonická struktura) a funkcí Matlabu `fft` – nezapomeňte zobrazovat vždy jen **modul**, který získáte pomocí `abs`. Ještě lepší bude zkusit zobrazit “plovoucí” spektrogram:

```
% ctení výsledku - spektrogram
ff = fopen('/tmp/specgram','r');
S = fread(ff, [256 inf], 'float');
S = S(1:129,:); % proc to delame ? a proc ne jen 1:128 ?
% abychom dostali plot stejný jako Matlab, musíme zobrazit v logaritmu [dB] a otocit osy
subplot(311); plot(s); axis tight;
subplot(312); imagesc(20*log10(S)); axis xy
% Matlab
subplot(313); specgram(s,256,8000,boxcar(240),160);
```

Úkoly

1. Co znamenají jednotlivé parametry u Matlab-funkce `specgram`? Prostudujte její help!
2. Ověřte, zda jsou stejné nejen obrázky, ale i numerické hodnoty (uložení výsledku `specgram` do matice).
3. Zkuste v Matlabu vygenerovat nějaký signál se známým chováním (např. kosinusovku na 1 kHz), uložit jej, vypočítat spektrogram a ten zobrazit. Je na něm zhruba to, co jste očekávali?