VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

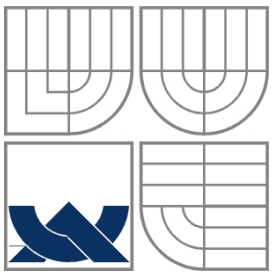
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVAČ HUDEBNÍHO STYLU Z MP3

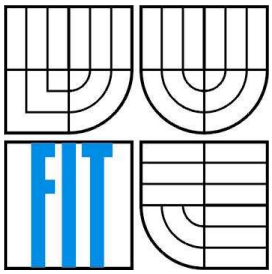
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE LUBOŠ DUCHOŇ
AUTHOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVAČ HUDEBNÍHO STYLU Z MP3

MUSIC STYLE RECOGNIZER FROM MP3

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — LUBOŠ DUCHOŇ
AUTHOR

VEDOUCÍ PRÁCE — Ing. FRANTIŠEK GRÉZL, Ph.D.
SUPERVISOR

BRNO 2009

Abstrakt

Tato bakalářská práce se zabývá detailním popisem zvukového formátu MP3 a návrhem rozpoznávače hudebních stylů z MP3 souborů, založeném na rozpoznávání pomocí skrytých Markovových modelů a koeficientů získaných přímo z MP3 souborů, s využitím nástrojů HTK.

Abstract

This bachelor’s thesis deals with detailed description of MP3 audio data format and music style recognizer. This recognizer is based on HTK Hidden Markov Models toolkit and coefficients obtained directly from MP3 files.

Klíčová slova

MP3, rozpoznávač, hudební styl, klasifikace, kódování, transformace, HTK, Mel-frekvenční kepstrální koeficienty, skryté Markovovy modely

Keywords

MP3, recognizer, music style, classification, coding, transformation, HTK, Mel-Frequency Cepstral Coefficients, Hidden Markov Models

Citace

Duchoň Luboš: Rozpoznávač hudebního stylu z MP3, bakalářská práce, Brno, FIT VUT v Brně, 2009

Rozpoznávač hudebního stylu z MP3

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Františka Grézla, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Luboš Duchoň
3. května 2009

Poděkování

Velmi rád bych tímto poděkoval vedoucímu práce, Ing. Františku Grézlovi, Ph.D., za ochotu, vstřícnost, odborné rady a pomoc při tvorbě této práce.

Obsah

1 Úvod

Cílem této bakalářské práce je seznámit se s metodami používanými pro klasifikování hudebních stylů, detailněji se seznámit se zvukovým formátem MP3 a navrhnout metodu rozpoznávání hudebních stylů z audia komprimovaného do formátu MP3.

1.1 Proč klasifikovat?

Řekne-li nám někdo *„Poslouchal jsem hudbu...“*, jen těžko si budeme představovat, o jakém stylu hudby je řeč, zvlášť pokud ani nepadne zmínka např. o interpretovi nebo názvu skladby. Avšak samotný interpret nebo název skladby nemusí naplnit naše představy o hudebním stylu, jestliže jej neznáme.

Pokud zazní věta *„Poslouchám techno...“* nebo *„Oni hrají jazz...“*, máme ve většině případů ohledně stylu hudby dokonalou představu. Klasifikace hudebních stylů tedy velmi výrazně usnadňuje orientaci ve světě hudby.

Dále existují rozsáhlé hudební archivy, které by se bez řádného roztřídění nahrávek podle různých hudebních žánrů rozhodně neobešly.

V neposlední řadě je klasifikování základním kamenem pro nejrůznější systémy, vyhledávající a nabízející uživatelům alternativní hudební nahrávky ze stejného žánru.

1.2 Metody rozpoznávání stylů

Většina metod, používaných pro rozpoznávání, je založena na principech strojového učení získaných vlastností pomocí rámcového extrahování.

Rámcová extrakce spočívá v rozdělení signálu do jednotlivých rámců a pro každý rámec se vypočítá vektor vlastností nějakého nízkoúrovňového popisovače, např. barva tónu (timbre) nebo rytmus. Hodnoty mohou být získány pomocí transformací, např. *FFT* nebo *MFCC koeficienty*, pomocí lineárního filtru, např. metodou *lineární predikce* nebo porovnáváním jednotlivých rámců, např. metoda *Spektrálního střediska* (Spectral centroid) *nebo Spektrálního toku* (Spectral flux).

Strojové učení je založeno na vytvořených modelech (*Skryté Markovovy modely*, *Neuronové sítě*) a algoritmech pro strojové učení (*Gaussův model*, *Linerární a nelineární klasifikátor*, *Vektorová kvantizace*).

Práce [5] popisuje a srovnává rozpoznávání hudebních stylů pomocí Bayesova klasifikátoru lineárního klasifikátoru a neuronových sítí. Bylo navrženo celkem 8 různých stylů a každý styl obsahoval 25 vzorků. Výsledky se lišily v závislosti na počtu rozpoznávaných stylů. Procentuální úspěšnost rozpoznávání je uvedeno v tabulce 1.

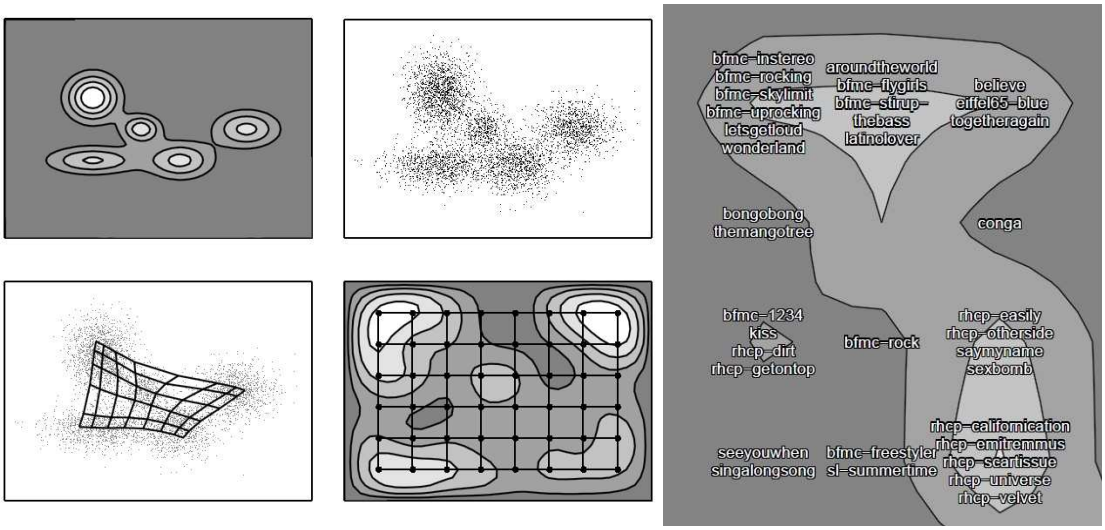
Počet stylů	Bayesův klasifikátor	Lineární klasifikátor	Neuronové sítě
4	98.1%	99.4%	98.5%
8	90.0%	84.3%	77.0%

Tabulka 1: Procentuální úspěšnost jednotlivých klasifikátorů

Velmi vhodné jsou pro rozpoznávání a třídění také *dynamické vlastnosti* hudby v příslušných frekvenčních pásmech. Patří sem zejména *rytmika*, tedy většinou charakteristika linky bicích nástrojů a perkusí, která značí, jak rychlé a silné jsou údery na jednotlivé doby a *výrazové a formotvorné prvky*, tedy nejrůznější sóla, přechody nebo přírazy.

Dalším způsobem, jakým se dá klasifikovat hudební styl, je zaměřit se na *charakteristické projevy* v jednotlivých žánrech. Např. pro rockové styly je charakteristická dominance elektrické kytary se zkreslením typu „overdrive" nebo „distortion", zatímco pro styl elektronické hudby jsou dominantní výrazné basy a syntezátorové zvuky nejčastěji typu „sawtooth", s nejrůznějšími typy zkreslení a úrovněmi jasu.

Dynamickými vlastnostmi a charakteristickými projevy použitými pro rozpoznávání hudebních stylů se zabývá práce [9]. Metoda byla nazvána „*Hudební ostrovy*" (Islands of Music) s využitím tzv. „samoorganizovaných map" (Self-Organizing Map) a bylo použito 359 různých hudebních vzorků. Nebyly však definovány jednotlivé styly, šlo pouze o srovnávání jednotlivých vzorků na základě již zmíněných vlastností a „podobné" vzorky se umísťovaly do stejné oblasti (ostrov).



Obrázek 1: Demonstrace SOM v projektu „Hudební ostrovy" [9]

2 MPEG-1 Audio Layer 3

Zřejmě heslem *„Kdo šetří, má za hodně...“* se řídili tvůrci při vývoji formátu MP3, který se stal jedním z nejpoužívanějších formátů pro uchovávání a přehrávání zvukových souborů a jejich sdílení v počítačových sítích. Velikost MP3 souboru je typicky oproti originálnímu zvukovému zdroji (např. WAV soubor) desetkrát menší, což rozhodně není zanedbatelné a navíc se v maximální možné míře zachovává kvalita původního zvukového zdroje.

2.1 Princip kódování

Formát MP3, oficiálním názvem *Moving Picture Experts Group-1 Audio Layer 3*, je založen na principu ztrátové komprese dat. Překódovat zvukový soubor např. z formátu WAV do formátu MP3 rozhodně není triviální záležitost, jak by se mohlo na první pohled zdát a celý proces sestává z několika etap popsaných níže.

Informace jsem čerpal z [2], [3], [7], [8], [11], [12], [16], [17], [18] a [19].

2.1.1 Analýza polyfázové filtrovací banky

Filtrovací banka rozdělí vstupní signál do 32 prostorově stejných frekvenčních podskupin.
V jednom rámci je filtrována sekvence 1152 PCM vzorků a každá podskupina tedy obsahuje celkem 36 vzorků.

$$St[i] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] * (C[k + 64j] * x[k + 64j]) \qquad (1)$$

i je index podskupiny s rozsahem 0 – 31.
St[i] je výstupní vzorek filtru pro skupinu *i* v čase *t*, kde *t* je celočíselný násobek 32 intervalů audio vzorku.
C[n] je jeden z 512 koeficientů okna analýzy definované ve standardu.
X[n] je vzorek vstupního zvuku, čteného z 512 vzorkového bufferu.
$M[i][k] = \cos[\frac{(2*i+1)*(k-16)*\pi}{64}]$ jsou koeficienty matice analýzy.

Mnohem výhodnější je však použití vzorce:

$$St[i] = \sum_{n=0}^{511} x[t - n] * Hi[n] \qquad (2)$$

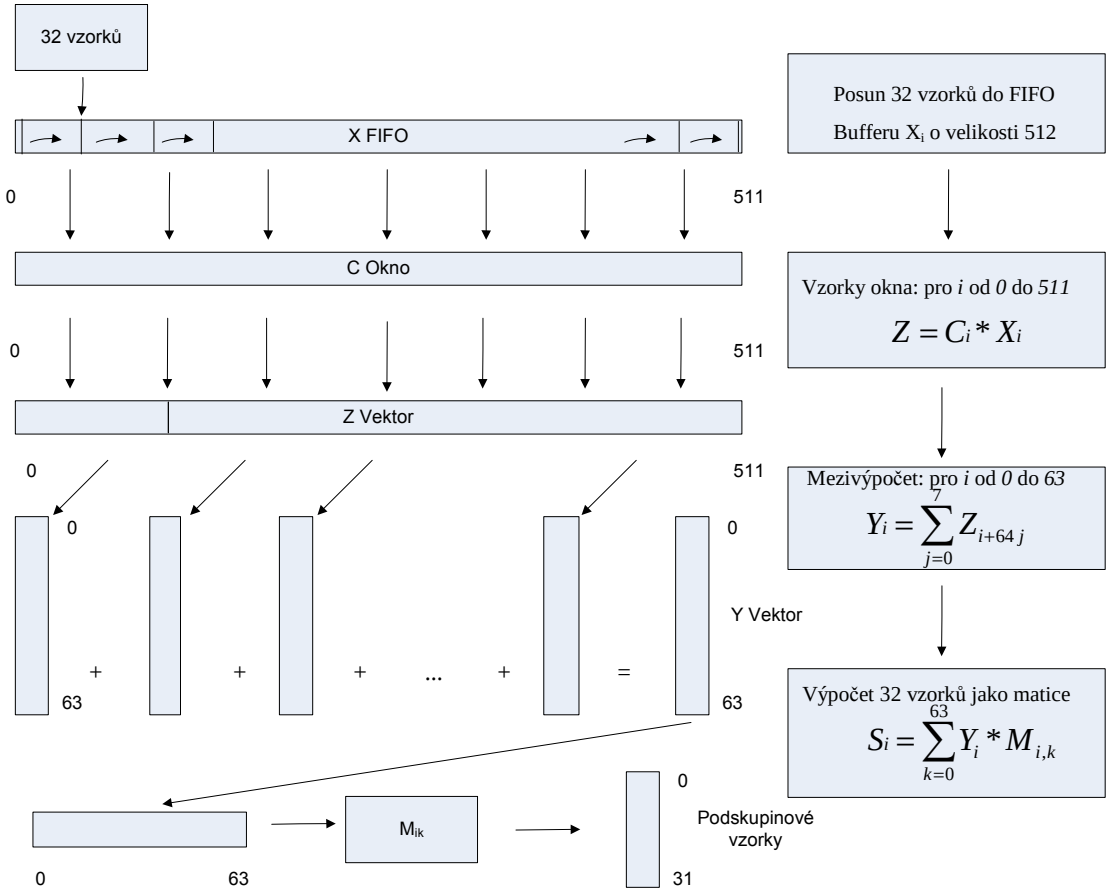
$x[\tau]$ je audio vzorek v čase τ

$$Hi[n] = h[n] * \cos[\frac{(2*i+1)*(n-16)*\pi}{64}] \qquad (3)$$

h[n] = *-C[n]* pokud je celočíselná část *n/64* lichá, jinak *h[n] = C[n]* pro *n* v rozsahu 0 – 511.

Koeficienty *h[n]* jsou prototyp filtru s dolní propustí pro filtrovací banku. Modulací prototypového filtru *h[n]* pomocí kosinu dojde k posunutí filtru.
H[i] jsou filtrovací banky, které posouvají odezvu dolní propusti k odpovídající frekvenční skupině, proto se nazývají polyfázové.

Tyto filtry mají střední frekvence na lichých násobcích $\frac{\pi}{64T}$ a každá má frekvenční rozsah $\frac{\pi}{32T}$, kde *T* je perioda vzorkování. Proces analýzy filtrovací banky ilustruje diagram na obrázku 2.



Obrázek 2: Diagram a procedura analýzy polyfázové filtrovací banky [7]

2.1.2 Modifikovaná diskrétní kosinová transformace

Výsledkem tohoto procesu je 32 podskupin reprezentováno *modifikovanou diskrétní kosinovou transformací* (MDKT). Tím se dosáhne zlepšení frekvenčního rozlišení u každé podskupiny.
Vzorec pro výpočet modifikované diskrétní kosinová transformace je následující:

$$X_i = \sum_{k=0}^{n-1} Z_k \cos[\frac{\pi}{2n}(2k + 1 + \frac{n}{2})(2i + 1)], \text{ pro } i = 0 \sim \frac{n}{2} - 1 \qquad (4)$$

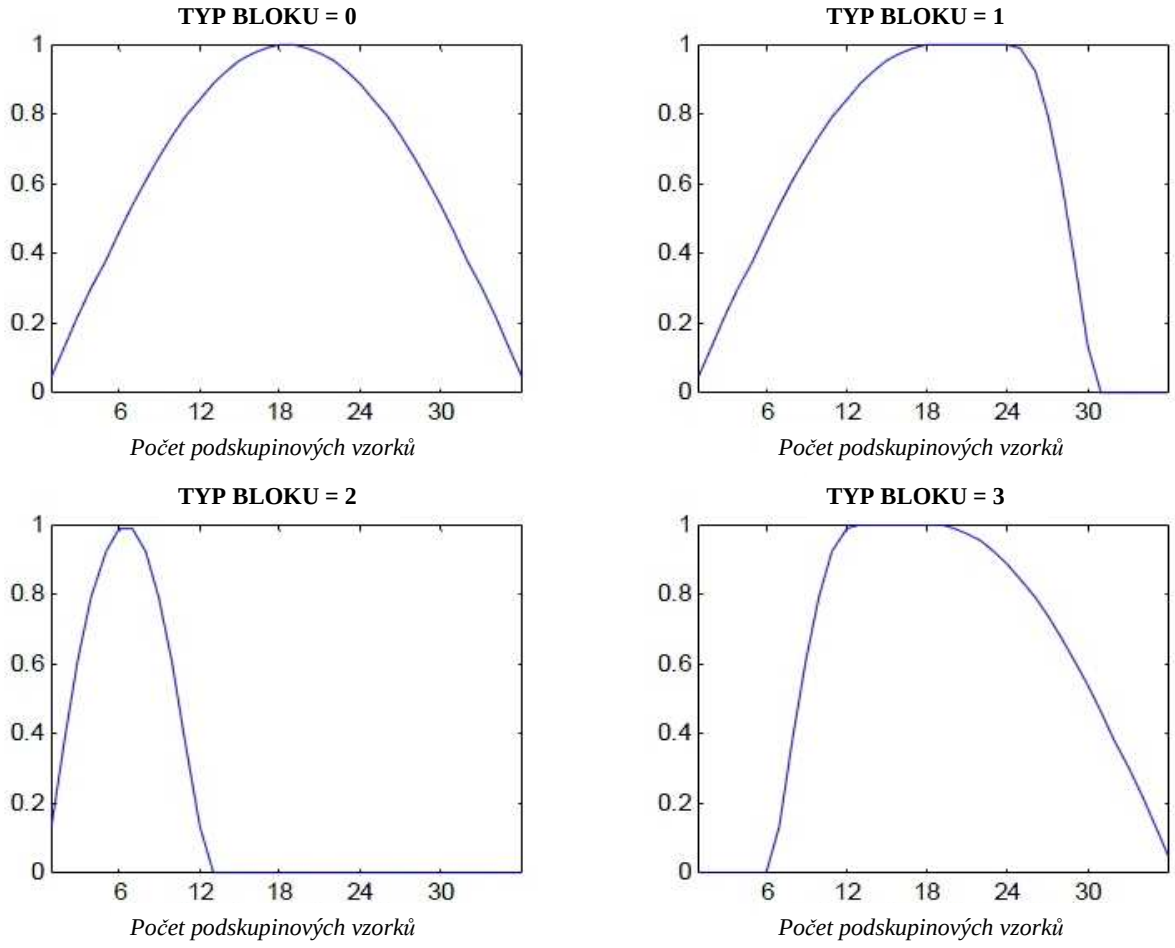
Před výpočtem transformace se na 36 podskupinových vzorků aplikují tzv. „*okenní*“ funkce, které usnadňují přechody mezi jednotlivými poskupinami a navíc zlepšují časové a frekvenční rozlišení. Charakteristiku těchto funkcí ilustruje obrázek 3.

Typ bloku = 0 – ***dlouhé okno***: $z_i = x_i \sin[\frac{\pi}{36}(i + \frac{1}{2})]$ pro $i = 0 \sim 35$ (5)

Typ bloku = 1 – ***start okno***: $$z_i = \begin{cases} x_i \sin[\frac{\pi}{36}(i + \frac{1}{2})] & i = 0 \sim 17 \\ x_i & i = 18 \sim 23 \\ x_i \sin[\frac{\pi}{12}(i - 18 + \frac{1}{2})] & i = 24 \sim 29 \\ 0 & i = 30 \sim 35 \end{cases} \text{ pro} \qquad (6)$$

Typ bloku = 2 – ***krátké okno***: $z_i = x_i \sin[\frac{\pi}{12}(i + \frac{1}{2})]$ pro $i = 0 \sim 11$ (7)

Typ bloku = 3 – ***stop okno***: $$z_i = \begin{cases} 0 & i = 0 \sim 5 \\ x_i \sin[\frac{\pi}{36}(i + \frac{1}{2})] & i = 6 \sim 11 \\ x_i & i = 12 \sim 17 \\ x_i \sin[\frac{\pi}{12}(i - 18 + \frac{1}{2})] & i = 18 \sim 35 \end{cases} \text{ pro} \qquad (8)$$



Obrázek 3: Charakteristika okenních funkcí [7]

Formát MP3 specifikuje 2 různé délky bloků MDKT – *dlouhý blok*, obsahující 18 vzorků a *krátký blok*, obsahující 6 vzorků.

Dlouhé nebo krátké okno se používá v závislosti na dynamice v každé podskupině. Pokud vzorky v dané podskupině vykazují stacionární chování, je použito *dlouhé okno* (typ 0). Pokud vzorky mění chování, je použito *krátké okno* (typ 2).

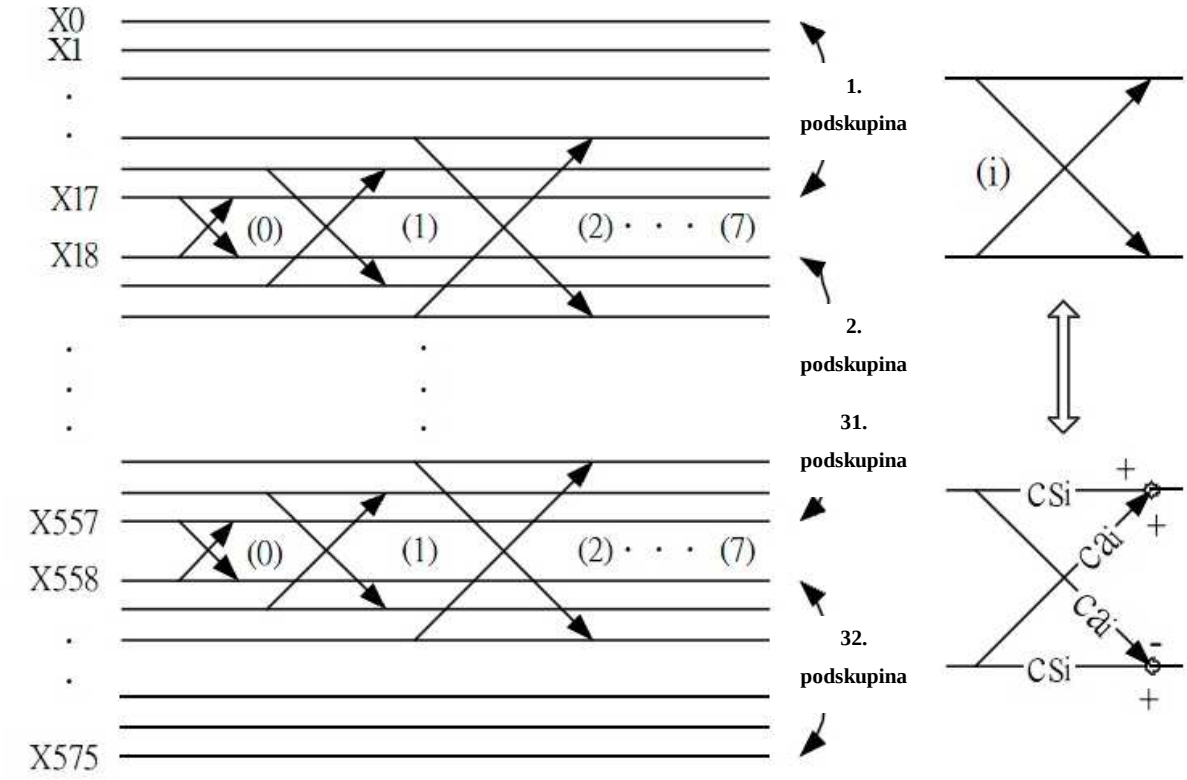
Zbývající dvě okna se používají pro ovládání přechodů z dlouhého na krátké okno a opačně, nazývají se *start okno* (typ 1) a *stop okno* (typ 3).

MDKT používá dlouhé okno pro 2 nízkofrekvenční podskupiny a krátké okno pro vyšších 30 podskupin. Tento způsob dává lepší frekvenční rozlišení pro nižší frekvence.

2.1.3 Redukce aliasu

Aliasing, který vznikl polyfázovou filtrovací bankou, se odstraní, aby se redukovalo množství informací, potřebných k přenosu. Používá se série *motýlkových výpočtů*, které přičítají vážené, zrcadlové verze sousedících podskupin sobě navzájem. Princip motýlkových výpočtů ilustruje obrázek 4.

Konstanty cs_i a ca_i jsou definované ve standardu ISO/IEC 11172-3:1993.



Obrázek 4: Redukce aliasu pomocí motýlkových výpočtů [7]

2.1.4 Psychoakustický model

Psychoakustický model je vzorek, který simuluje lidské vnímání zvuku. Tento model je použit pouze pro rozhodování, které části audio signálu jsou akusticky irelevantní a které nejsou a odstraňuje neslyšitelné části.

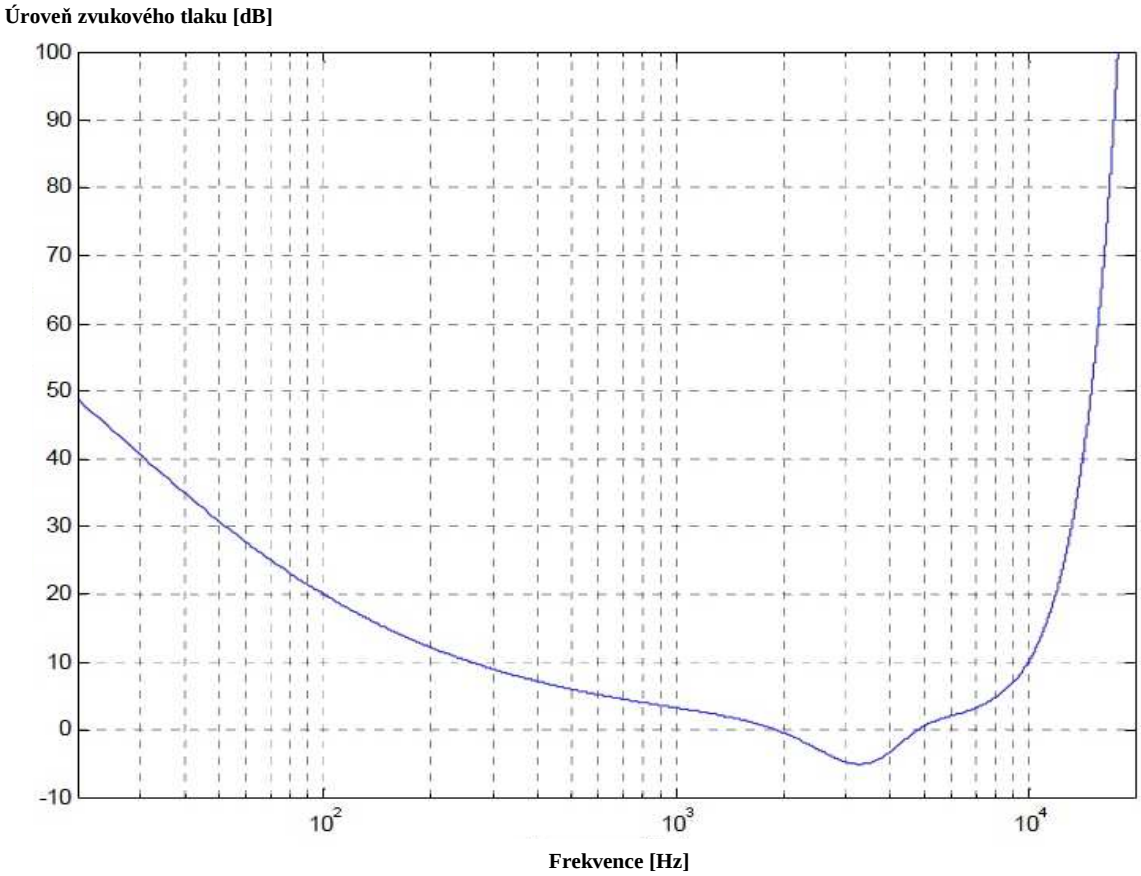
Psychoakustický model poskytuje neuniformnímu kvantizačnímu bloku informace, jak kvantovat frekveční linie. Kvantování frekvenčních linek je přizpůsobeno limitům vnímání lidského ucha.

Psychoakustický model sestává ze 3 maskovacích principů:

ABSOLUTNÍ PRÁCH SLYŠITELNOSTI

Absolutní prách slyšitelnosti je charakterizován minimálním množstvím energie, potřebné v čistém tónu, aby bylo možné jej detekovat posluchačem v tichém prostředí, přičemž posluchač nemá žádné znalosti o aktuálních úrovních přehrávání.

Hodnoty energie – *úroveň zvukového tlaku* (SPL – Sound Pressure Level) jsou vyjádřeny v decibelech. Charakteristiku absolutního prahu slyšitelnosti ilustruje Obrázek 5.



Obrázek 5: Absolutní prách slyšitelnosti [7]

FREKVENČNÍ MASKOVÁNÍ

Pomocí frekvečního maskování se signál nízké úrovně převádí na neslyšitelný s pomocí současně se vyskytujícího silnějšího signálu tak dlouho, dokud se oba příliš frekvenčně neliší.

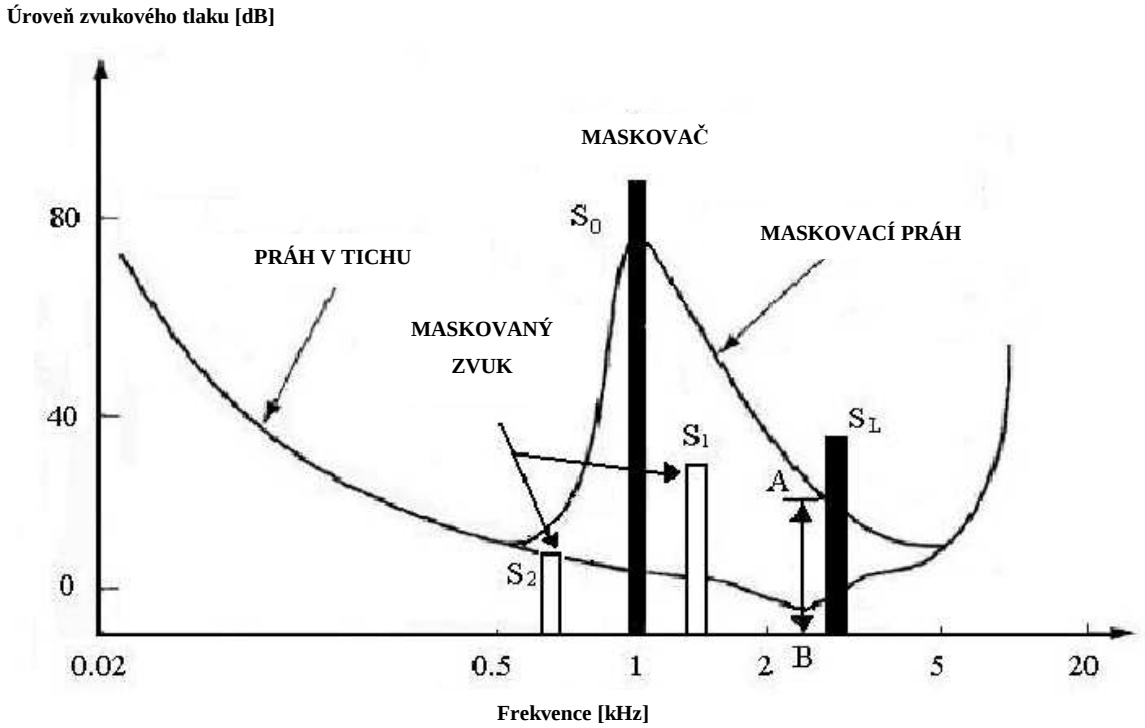
Maskovací práh závisí na úrovni zvukového tlaku a na frekvenci silnějšího signálu. Frekvence maskovacího prahu může být posouvána, dokud nebude žádný signál slyšitelný.

Princip maskování ilustruje obrázek 6.

Nejsilnějším signálem je signál S_0. Energie signálu pod hranicí maskovacího prahu bude maskována za pomocí S_0. Slabší signály S_1 a S_2 nejsou slyšitelné, protože jejich úroveň zvukového tlaku je pod maskovacím prahem. Signál S_L je maskován částečně.

Je tedy možné zvýšit kvantizační šum v podskupině, obsahující signál S_L o úroveň AB, což znamená, že pro reprezentaci signálu v této podskupině je potřeba méně bitů.

Bez využití maskovacího signálu také není signál slyšitelný, pokud je úroveň tlaku zvuku pod absolutním prahem.



Obrázek 6: Frekvenční maskování [7]

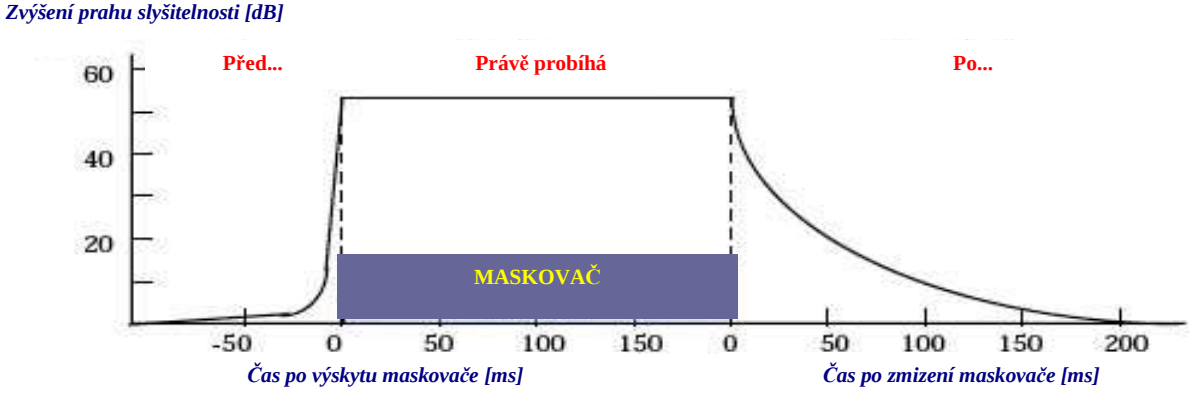
ČASOVÉ MASKOVÁNÍ

Časové maskování hraje důležitou roli pro lidské zvukové vnímání v čase.

Maskování může nastat, pokud se dva zvuky objeví v krátkém časovém intervalu. Silnější zvuk může maskovat slabší i tehdy, pokud slabší signál předchází silnějšímu.

Maskování nastává i při zániku maskovacího signálu a trvá určitou dobu, než je lidské ucho schopné vnímat současně působící slabší signál.

Obrázek 7 ilustruje průběh maskování ve 3 etapách (doba před výskytem maskovače, doba maskování a doba zániku maskovače):



Obrázek 7: Časové maskování [2]

2.1.5 Huffmanovo kódování

V tomto bloku se provádí kódování kvantovaných frekvenčních linií pomocí algoritmu Huffmanova kódování, založeného na tzv. 32 statických Huffmanových tabulkách.

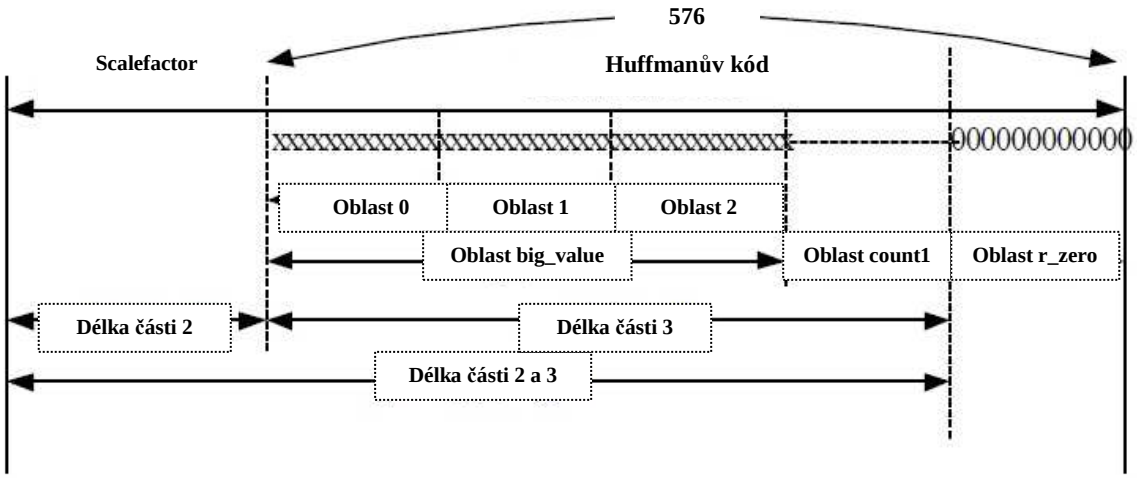
Huffmanovo kódování provádí ztrátovou kompresi, čímž redukuje množství přenášených dat, aniž by docházelo ke ztrátě kvality. MP3 stanovuje frekvenční linie do 3 oblastí, nazývaných **rzero**, **count1** a **big_value**. Struktura je ilustrována na obrázku 8.

Začíná se na vyšších frekvencích a identifikuje se kontinuální blok všech nulových hodnot jako *rzero*. Tato oblast nebude kódována, protože její velikost může být odvozena od velikosti dalších 2 oblastí.

Oblast *count1_region* zahrnuje kontinuální blok hodnot -1, 0 nebo 1. Pro tuto oblast kódují 2 Huffmanovy tabulky v jednom okamžiku 4 hodnoty, takže počet hodnot v této oblasti musí být násobkem 4.

Oblast *big_values region* zahrnuje všechny zbývající hodnoty. 30 Huffmanových tabulek pro tuto oblast kóduje hodnoty ve dvojici. Oblast je dále rozdělena do 3 podoblastí a každá má vlastní Huffmanovu tabulku. Do této oblasti je zavedena ESCAPE hodnota, která zvyšuje efektivnost kódování ještě před kódováním frekvenčních linií. Hodnoty překračující 15 jsou reprezentovány číslem 15 a zbytek je ESCAPE hodnota. V závislosti na velikosti ESCAPE hodnoty se stanovuje počet bitů nazývaných „*linbits*“, které reprezentují ESCAPE hodnotu.

Hodnoty Huffmanových tabulek jsou uvedeny v tabulce A1 přílohy A.



Obrázek 8: Huffmanovo kódování [7]

2.1.6 Formátování datového toku

Poslední blok kódovacího procesu produkuje výsledný datový tok MP3 souboru. Ten se skládá z frekvenčních linií kódovaných Huffmanovým kódováním, postranních informací a záhlaví rámce.

Datový tok je rozdělen na rámce, kde každý obsahuje 1152 zvukových vzorků. Záhlaví popisuje rychlost datového toku a vzorkovací frekvenci. Postranní informace udává typ bloku, Huffmanovy tabulky a vybrané podskupinové faktory a zisky.

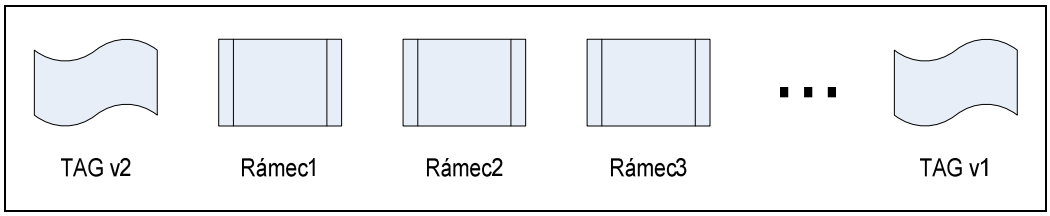
2.1.7 Bitový zásobník

Bitový zásobník se používá pro uspořádání časově-variabilní závislosti na bitech kódu. Během kódování se do zásobníku mohou vložit bity, pokud je potřeba méně bitů, než je průměrný počet bitů pro kódování rámce. Pokud je potřeba více bitů, vyberou se ze zásobníku. Ze zásobníku je možné vybírat pouze bity uložené z předešlých rámců.

Formát MP3 používá 9-bitový ukazatel nazývaný ***main_data_begin***, nachází se v postranních informacích každého rámce a ukazuje na místo počátečního bytu audio dat rámce.

2.2 Struktura souboru MP3

Soubor ve formátu MP3 je rozdělen do tzv. *rámců* (frames) o konstatní časové délce 0.026 sekund.



Obrázek 9: Struktura souboru MP3

Velikost rámce v bytech závisí na *přenosové rychlosti* (bitrate). První 4 byty každého rámce je tzv. *záhlaví rámce* (frame header), dále pak následuje 17 nebo 32 bytů (záleží na tom, zda je použit jeden nebo dva kanály) tzv. *postranních informací* a zbytek tvoří audio data.

2.2.1 Struktura záhlaví rámce

Záhlaví rámce má následující strukturu, každé písmeno reprezentuje 1 bit. Detailní popis jednotlivých částí je uveden v tabulkách B1 – B11 přílohy B.

A A A A A A A A A A A B B C C D E E E E F F G H I I J J K L M M

- **A**: ***Frame synchronizer*** ***(11 bitů)***

Všechny bity jsou zde nastaveny na hodnotu 1, používá se pro nalezení začátku rámce.

- **B**: ***MPEG Version ID*** ***(2 bity)***
- **C**: ***Layer*** ***(2 bity)***
- **D**: ***CRC Protection*** ***(1 bit)***
- **E**: ***Bitrate index*** ***(4 bity)***
- **F**: ***Sampling rate frequency index*** ***(2 bity)***
- **G**: ***Padding*** ***(1 bit)***
- **H**: ***Private bit*** ***(1 bit)***
- **I**: ***Channel mode*** ***(2 bity)***
- **J**: ***Mode extension*** *(pouze pro Joint Stereo)* ***(2 bity)***
- **K**: ***Copyright*** ***(1 bit)***
- **L**: ***Original*** ***(1 bit)***
- **M**: ***Emphasis*** ***(2 bity)***

Pro výpočet délky rámce v bytech se používá vztah:

$$FrameLen = \text{int}((144 * BitRate / SampleRate) + Padding) \quad (9)$$

int() znamená zaokrouhlení směrem dolů.

Příklad:
Pro datový tok 192 kbit/s, vzorkovací frekvenci 44100 Hz a nastavený příznak Padding, je délka rámce $FrameLen = \text{int}((144 * 192000 / 44100) + 1) = \text{int}(627,938) = 627$ bytů.

2.2.2 Struktura bloku postranních informací

Zde jsou uloženy informace potřebné k dekódování audio dat. Velikost těchto informací závisí na počtu použitých kanálů. Pro jeden kanál je velikost 17 bytů (136 bitů) a pro dva kanály je velikost 32 bytů (256 bitů). Celek se skládá z 5 částí:

Main data begin	Private bits	Scfsi	Granule 0 Side info	Granule 1 Side info

Main data begin (9 bitů)
Použitím techniky *bitového zásobníku* je umožněno využít zbývající prázdné místo po sobě jdoucími rámci. Tato hodnota přesně udává začátek hlavních dat v každém rámci. Pokud je hodnota 0, začátek dat je přímo po bloku postranních informací.

Private bits (5 bitů)
Jedná se o bity pro individuální použití. Tato hodnota však nebude v budoucnu součástí ISO normy.

Scfsi (4 bity / 8 bitů)
Scfsi (Scale factor selection information) udává, zda jsou stejná měřítka přenášena pro obě tzv. *„zrnka“* (granule). Soubory měřítek jsou rozděleny do 4 kategorií:

Granule Side information (59 bitů / 118 bitů)

Tento blok je rozdělen do 14 částí. Detailní hodnoty jsou uvedeny v tabulkách C1 – C3 přílohy C.

part2_3_length
big_values
global_gain
scalefac_compress
windows_switching_flag
block_type
mixed_block_flag
table_select
subblock_gain
region0_count
region1_count
preflag
scalefac_scale
count1table_select

Part2_3_length (12 bitů / 24 bitů)
Udává počet bitů alokovaných v části hlavních dat rámce pro měřítka (part2) a data kódována pomocí Huffmanova kódování (part3). Používá se pro výpočet pozice dalšího zrnka a doplňkových informací (pokud jsou použity).

Big_values (9 bitů / 18 bitů)
Udává hodnoty používané pro kódování spektrálních oblastí pomocí různých Huffmanových tabulek.

Global_gain (8 bitů / 16 bitů)
Specifikuje rozsah kvantizačního kroku. Je potřebný pro rekvantizační blok dekodéru.

Scalefac_compress (4 bity / 8 bitů)
Udává počet bitů použitých pro přenos měřítek. V závislosti na použitých okenních funkcích se zrnko dělí do souboru 21 měřítek (dlouhé okno, použití typu bloku 0, 1 nebo 3) nebo 12 měřítek (krátké okno, použití typu bloku 2).

Měřítka se dále rozdělí do 2 kategorií, opět v závislosti na použitých okenních funkcích. Pro dlouhé okno je kategorie 0 – 10 a 11 – 20, pro krátké okno je kategorie 0 – 6 a 7 – 11.

Počet bitů pro každou skupinu uvádějí hodnoty *slen1* a *slen2* v následující tabulce:

Windows_switching_flag (4 bity / 8 bitů)
Příznak udávající použití jiného, než standardního okna. Pokud je tento příznak nastaven, používají se bloky *block_type*, *mixed_block_flag* a *subblock_gain*.

Block_type (2 bity / 4 bity)
Indikuje typ použitého okna. Hodnota 00 není povolena, pokud se používá jiný než standardní typ okna.

Mixed_block_flag (1 bit / 2 bity)
Indikuje použití jiných typů okna u nižších a vyšších frekvencí. 2 nejnižší podskupiny jsou transformovány s použitím standardního typu okna a zbývajících 30 podskupin je transformováno použitím jiného typu okna definovaného v *block_type*.

Table_select (10 bitů / 20 bitů nebo 15 bitů / 30 bitů)
Tato hodnota udává typ použité Huffmanovy tabulky pro dekódování oblasti *big_value*. Velikost indexu Huffmanovy tabulky je 5 bitů. Pokud je nastaven příznak *windows_switching_flag*, oblast *region2* je prázná a kódují se tedy pouze 2 oblasti, je celková velikost 10 bitů (*5*2*1)* pro mono a 20 bitů (*5*2*2*) pro stereo.

Pokud není nastaven příznak *windows_switching_flag*, kódují se všechny 3 oblasti a celková velikost je tedy 15 bitů (*5*3*1*) pro mono a 30 bitů (*5*3*2*) pro stereo.

Subblock_gain (9 bitů / 18 bitů)
Udává offset zisku (offset gain) z globálního zisku (global gain) pro každé krátké okno. Celkem se tedy jedná o 9 bitů (3 bity pro každé okno, je-li nastaven příznak *windows_switching_flag* a hodnota *block_type* je 10).

Region0_count (4 bity / 8 bitů,) Region1_count (3 bity / 6 bitů)
Obsahuje hodnotu o jedničku menší, než je uvedený počet skupin měřítek u oblastí *region0* a *region1*. Hranice oblastí jsou uzpůsobeny pro členění frekvenčního spektra do skupin měřítek. Pokud se používají krátká okna, sčítá se počet jednotlivých oken.
Příklad:

Hodnota oblasti *region0* je 8, obsahuje tedy 3 skupiny měřítek $\left(\frac{9}{3}\right)$.

Preflag (1 bit / 2 bity)
Používá se pro přídavné zesílení kvantovaných hodnot vysokých frekvencí. Pro krátká okna se nepoužívá.

Scalefac_scale (1 bit / 2 bity)

Pro hodnotu 0 jsou měřítka logaritmicky kvantována s krokem 2. Pro hodnotu 1 s krokem $\sqrt{2}$.

Count1table_select (1 bit / 2 bity)
Udává použití Huffmanovy tabulky pro oblast *count1*.

2.2.3 Struktura záhlaví rámce s variabilním datovým tokem

Systém variabilního datového toku byl vytvořen za účelem snížení velikosti souboru a zachování zvukové kvality.

Principem je užívání vyššího nebo nižšího datového toku v závislosti na frekvencích tónů. Vyšší tóny potřebují více prostoru pro kódování, zatímco u nižších tónů je tomu naopak. Pokud se v některých částech vyšší tóny nevyskytují, je zbytečné používat vyšší datový tok (např. 192 kbit/s), což je mnohem výhodnější z hlediska prostoru a naprosto dostačující z hlediska kvality použít pouze datový tok např. 96 kbit/s.

Po řetězci „*Xing*“ následují příznaky, počet rámců v souboru a velikost souboru v bytech. Všechny tyto hodnoty mají velikost 4 byty.

Detailní popis a hodnoty jsou uvedeny v tabulkách D1 – D3 přílohy D.

Frames Flag je nastaven, pokud je uložena hodnota o počtu rámců v souboru.
Bytes Flag je nastaven, pokud je uložena hodnota o velikosti souboru v bytech.
Table of Contents (TOC) Flag je nastaven, pokud jsou uloženy hodnoty pro obsah.
Variable Bit Rate (VBR) Flag je nastaven, pokud jsou uloženy hodnoty pro rozsah variabilního datového toku.

2.2.4 TAGy

Jedná se o datové prostory v MP3, kam se mohou ukládat textové informace, např. název skladby, interpret, album, žánr, ...).

TAG ID3 verze 1
Je starší a jednodušší. Jeho velikost je vždy 128 bytů a je umístěn za posledním audio rámci na konci souboru.

Velkou nevýhodou TAGu ID3 verze 1 je jeho omezená velikost. Je možné uložit pouze několik položek, pro které je limitovaný prostor.

Není tedy možné uložit například informaci o zemi původu, použitém kodeku nebo není možné uložit například název skladby delší než 30 znaků.

Struktura TAGu ID3 verze 1 je uveden v tabulce E1 přílohy E.

TAG ID3 verze 2
Je novější a větší, je umístěn před všemi audio rámci, na samém začátku audio souboru.
Obsahuje záhlaví a rámce.

ZÁHLAVÍ

Detailní popis záhlaví je uveden v tabulce E2 přílohy E.

Byte příznaků (Flags Byte) má následující strukturu:

A	B	C	0	0	0	0	0

První tři byty indikují *nesynchronizovanost* (unsynchronization), *přídavné záhlaví* (extended header) a *experimentální indikátor* (experimental indicator).

Příznaky nemají speciální význam a mohou být nastaveny na 00.

Velikost TAGu je kódována na 4 bytech. Nejvýznamější bit v každém bytu je nastaven na 0 a ignorován. Použito je jen zbývajících 7 bitů. Je to z důvodu ochrany před záměnou se záhlavím audio rámce, který má první synchonizační byte FF.

Příklad:
Velikost TAGu je 257, tzn., je kódována jako 00 00 01 01. Velikost TAGu neobsahuje vlastní záhlaví, celková velikost by tedy byla 257 + 10 bytů.

RÁMCE

Každý rámec ukládá jednu informaci, např. interpret nebo album. Rámec se sestává ze záhlaví a těla. Detailní popis rámce je uveden v tabulkách E3 a E4 přílohy E

Identifikátor rámce (Frame identifier)

Jedná se o čtyřznakový řetězec. Některé hodnoty jsou již předdefinovány a není problém vytvořit si vlastní identifikátor a tímto si do MP3 uložit jakoukoliv informaci, bez jakéhokoliv limitu.

Velikost (Size) je uložena od nejvíce významného bytu k nejméně významnému a nezahrnuje záhlaví rámce. Celková velikost je tedy velikost + 10. Po záhlaví následuje vždy byte s hodnotou 00 a poté začíná tělo rámce. Velikost musí zahrnovat i tento byte.

Příznaky (Flags) jsou ve většině případů nastaveny na hodnotu 00 00 a mají následující bitovou strukturu. Detailní popis je uveden v tabulce E5 přílohy E.

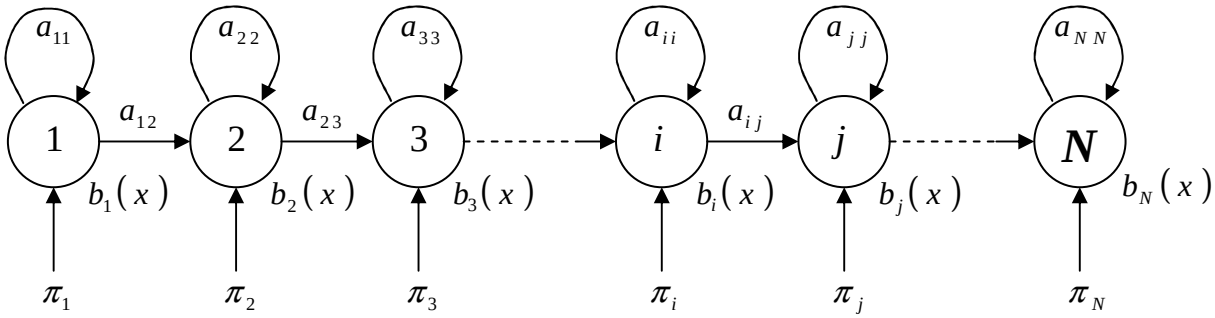
A	B	C	0	0	0	0	0	I	J	K	0	0	0	0	0

3 Návrh metod pro rozpoznávání

Pro vytvoření rozpoznávače využijeme metod skrytých Markovových modelů s pomocí nástrojů HTK, určených pro tvorbu těchto modelů a parametrizací pomocí MFCC koeficientů.

3.1 Skryté Markovovy modely

Skrytý Markovův model (HMM) je statistický model s konečným počtem stavů a s předpokladem, že se jedná o Markovský proces.



Obrázek 10: N-stavový skrytý Markovův model [6]

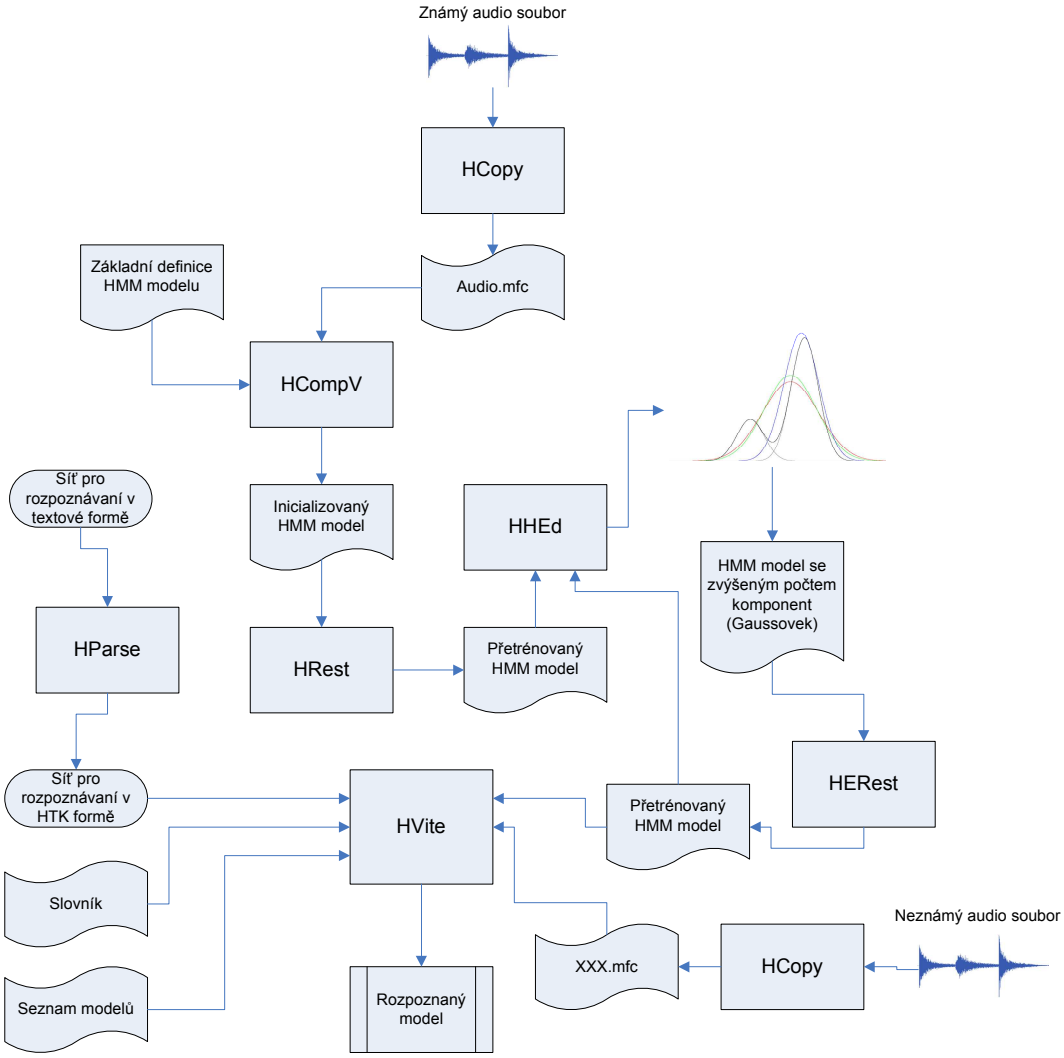
π_i - počáteční pravděpodobnost stavu
a_{ij} - pravděpodobnost přechodu mezi stavy *i* a *j*
$b_i(x)$ - hustota pravděpodobnosti pro pozorovaný vektor *x* v daném stavu
N - počet stavů
i, j - označení daných konkrétních stavů

3.2 Trénování modelů pomocí HTK

HTK je soubor nástrojů napsaných v jazyce C pro tvorbu skrytých Markovových modelů (HMM). Obsahuje nástroje pro trénování a rozpoznávání. Zjednodušené schéma činnosti této sady nástrojů je ilustrováno na obrázku 11.

Trénování spočívá v parametrizaci sady HMM za použití přepisů známých trénovacích vzorků. Rozpoznávání spočívá v přepisu neznámých vzorků a jejich rozpoznání pomocí rozpoznávacích nástrojů.

Pro práci s HTK nástroji je nutné vytvořit několik konfiguračních souborů, způsobem popsaným níže.



Obrázek 11: Zjednodušené schéma činnosti HTK

HCopy konvertuje vstupní data do parametrizované formy, nejčastěji se jedná o MFCC koeficienty.
HCompV spočítá globální význam a kovarianci sady trénovacích vzorků. Primárně se používá pro inicializaci parametrů HMM.
HRest provádí přetrénování parametrů HMM pomocí základního Baum-Welchova algoritmu.
HHEd editor HMM, pomocí příkazu MU je schopen zvýšit počet komponent procesem nazývaným „mixture splitting“ pro zvýšení flexibility.
HERest provádí přetrénování parametrů HMM pomocí Baum-Welchova algoritmu ve verzi vestavěného přetrénování.
HParse vygeneruje slovní rozpoznávací síť z textového souboru do sady pravidel založených na rozšířené Backus-Naurovy formě
HVite je rozpoznávač založený na Viterbiho algoritmu.

3.2.1 Extrahování parametrů

Sadu parametrů získáme pomocí nástroje *HCopy*. Pro tento nástroj je nutné vytvořit konfigurační soubor s určitými parametry. Budeme uvažovat parametry ve formě MFCC koeficientů bez energie a delta koeficientů, data budou typu WAVE a nebudou mít hlavičku.

```
SOURCEKIND          =   WAVEFORM  # typ dat bude WAVE
SOURCEFORMAT        =   NOHEAD    # data budou bez hlavičky
SOURCERATE          =   2500      # vzorkovací perioda zdroje ( 2500*100ns )
TARGETFORMAT        =   HTK       # výstupní formát bude HTK
TARGETKIND          =   MFCC_0    # MFCC koeficienty bez energie a delty
SAVEWITHCRC         =   FALSE     # bez kontrolního součtu
HPARM: NUMCHANS     =   40        # počet kritických skupin
HPARM: CEPLIFTER    =   30        # počet filtrovacích bank
HPARM: NUMCEPS      =   30        # počet výstupních parametrů
HPARM: USEPOWER     =   TRUE      # použití funkce síly spektra
HPARM: USEHAMMING   =   TRUE      # použití funkce Hamming window
HPARM: PREEMCOEF    =   0         # bez preemfáze
HPARM: TARGETRATE   =   100000    # vzorkovací perioda výstupu (10ms)
HPARM: WINDOWSIZE   =   250000    # délka okna (25ms)
```

3.2.2 Inicializace modelů

Základní inicializaci modelů provedeme nástrojem *HCompV*. Nejprve je nutné model definovat vytvořením prototypu.

Střední hodnoty (Mean) nastavíme na hodnotu 0.0 a hodnoty *rozptylu* (Variance) na 1.0, dále použijeme *3-stavový* (NumStates) skrytý Markovův model, kde 1. a 3. stav značí ticho:

```
~o <VecSize> 31 <MFCC_0>
~h "BEATOLDIES"
<BeginHMM>
<NumStates> 3
<State> 2 <NumMixes> 1
<Mean> 31
0.0 0.0 0.0 ...
<Variance> 31
1.0 1.0 1.0 ...
<TransP> 3
0.0 1.0 0.0
0.0 0.6 0.4
0.0 0.0 0.0
<EndHMM>
```

Pro nástroje *HCompV* a *HRest* je nutné vytvořit tzv. *Master-Label* soubory. Uvádí se cesta k label souboru, čas začátku a konce každého zvukového souboru je uvedený ve stovkách ns a návěští identické s názvem modelu.

```
#!MLF!#
"POP/BEATOLDIES/data/BEATOLDIES01.lab"
0       337885941       BEATOLDIES
.
"POP/BEATOLDIES/data/BEATOLDIES02.lab"
0       441149659       BEATOLDIES
.
"POP/BEATOLDIES/data/BEATOLDIES03.lab"
0       388246938       BEATOLDIES
.
atd...
```

Pro výpočet času konce souboru se použije vztah:

$$HTKčas = \frac{velikostSouboruVbytech}{200 \times vzorkovacíFrekvence} \times 10^{-9} \quad (14)$$

3.2.3 Trénování

Základní trénink provádíme pomocí nástroje *HRest*. Přetrénovaný model může vypadat např. takto:

```
~o
<STREAMINFO> 1 104
<VECSIZE> 104<NULLD><USER><DIAGC>
~h "BEATOLDIES"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 104
 1.247592e+02 1.247536e+02 1.247662e+02 ...
<VARIANCE> 104
 5.634122e+03 5.695348e+03 5.644566e+03 ...
<GCONST> 1.089810e+03
<TRANSP> 3
 0.000000e+00 1.000000e+00 0.000000e+00
 0.000000e+00 6.000000e-01 4.000000e-01
 0.000000e+00 0.000000e+00 0.000000e+00
<ENDHMM>
```

Pro dosažení co nejlepších výsledků, je však nutné průběžně zvyšovat počet komponent („Gaussovek“) tzv. „dělením směsí“ (mixture splitting) pomocí nástroje *HHEd*:

MU 24 {*.state[2].mix}

Poté se nový model se zvýšeným počtem komponent znovu přetrénuje pomocí nástroje *HERest*.

Dále je potřeba vytvořit síť pro rozpoznávání pro nástroj *HParse*, který převede tuto síť do formy pro HTK. Síť bude obsahovat jednotlivé hudební styly:

```
(
BEATOLDIES | POPBALLAD | RELAXDREAM | CONTEMPROCK | HARDMETAL | ROCKBALLAD | DISCOSCHLAGER | HIPHOPRAP | TRANCETECHNO | BIGBAND | COUNTRY | LATIN
)
```

Rozpoznávání provádíme nástrojem *HVite*. Pro tento nástroj je potřeba vytvořit seznam modelů hudebních stylů a slovník:

BEATOLDIES
POPBALLAD
RELAXDREAM
CONTEMPROCK
HARDMETAL
ROCKBALLAD
DISCOSCHLAGER
HIPHOPRAP
TRANCETECHNO
BIGBAND
COUNTRY
LATIN

BEATOLDIES	BEATOLDIES
POPBALLAD	POPBALLAD
RELAXDREAM	RELAXDREAM
CONTEMPROCK	CONTEMPROCK
HARDMETAL	HARDMETAL
ROCKBALLAD	ROCKBALLAD
DISCOSCHLAGER	DISCOSCHLAGER
HIPHOPRAP	HIPHOPRAP
TRANCETECHNO	TRANCETECHNO
BIGBAND	BIGBAND
COUNTRY	COUNTRY
LATIN	LATIN

4 Data a testování

4.1 Návrh rozpoznávaných stylů

V této práci jsou navrženy 4 skupiny hudebních stylů, každý ještě obsahuje 3 podskupiny, což činí celkem 12 různých hudebních stylů. Uvedené příklady skladeb jsou obsaženy v souboru nashromážděných trénovacích vzorků. Pro každý styl byla stanovena 1 hodina čisté hudby, což při průměrné délce skladby 3 minuty činí cca. 20 skladeb. Pro dosažení větší variability je každá skladba od jiného interpreta. Celkový počet trénovacích vzorků je 252 a celkově se jedná o cca. 16 hodin hudby. Pro samotné rozpoznávání bylo nashromážděno dalších 60 vzorků.

Seznam všech vzorků je uveden na přiloženém CD.

4.1.1 POP

8 Beat & Oldies
Styl zahrnuje klasické beatové skladby středního tempa (110 – 140 BPM) hlavně ze 60. a 70. let.
Příklady: Mike Oldfield: *Moonlight shadow*
Smokie: *Livin‘ next door to Alice*

Pop Ballad
Styl zahrnuje popové skladby s pomalým tempem (60 – 90 BPM) z 80. let až po současnost.
Příklady: Foreigner: *I want to know what love is*
Richard Marx: *Right here waiting*

Relax & Dream
Styl zahrnující relaxační a meditační hudbu.
Příklady: Enya: *Caribbean blue*
Vangelis: *Chariots of fire*

4.1.2 ROCK

Contemporary Rock & Big Beat
Styl zahrnující skladby klasické rockové scény současnosti i minulosti se středně rychlým tempem (130 – 160 BPM).
Příklady: Bryan Adams: *Summer of 69*
Kiss: *Lick it up*

Hard Rock & Metal
Styl zahrnující skladby tvrdé rockové scény současnosti i minulosti se středně rychlým až rychlým tempem (120 – 190 BPM).
Příklady: Dream Evil: *The Prophecy*
Rammstein: *Feuer frei*

Rock Ballad
Styl zahrnující klasické rockové balady s pomalým tempem (60 – 90 BPM).
Příklad: Gary Moore: *Still got the blues*
Guns N‘ Roses: *Knocking on heavens door*

4.1.3 DANCE

Disco & Schlager
Styl zahrnující klasické taneční skladby 70. až 90. let.
Příklad: Al Bano e Romina Power: *Mamma Maria*
Modern Talking: *You're my heart, you're my soul*

Trance & Techno
Styl zahrnující taneční skladby současné doby.
Příklad: DJ Sammy: *Heaven*
Kate Ryan: *Voyage voyage*

Hip Hop & Rap
Styl zahrnující hip hopové a rapové skladby.
Příklady: Kontrafakt: *Život je boj*
Usher: *Yeah*

4.1.4 BALLROOM

Big Band & Orchestra
Styl zahrnující klasické jazzové a swingové skladby z 30. – 50. let
Příklady: Billy Vaughn: *Sail along silvery moon*
Glenn Miller: *In the mood*

Latin

Styl zahrnující skladby latinsko-amerického stylu.
Příklady: Gipsy Kings: *Volare*
Kaoma: *Lambada*

Country
Styl zahrnující skladby klasického country a bluegrassu.
Příklady: Johnny Cash: *Ring of fire*
Michal Tučný: *Báječná ženská*

4.2 Parametrizace audio souborů

Parametrizace audio souborů se provede pomocí *MFCC koeficientů* (Mel-frequency cepstral coefficients), což je sada koeficientů tvořících tzv. *MFC kepstrum* (Mel-frequency cepstrum), které reprezentuje krátkodobé spektrum síly zvuku na nelineárním frekvenčním *Mel rozsahu*.

Signál se rozdělí do krátkých časových oken a u každého okna se provede diskrétní Fourierova transformace:

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)\exp(-j2\pi kn / N) \tag{10}$$

x(n) – diskrétní signál s délkou *N*

w(n) – časové okno vypočítané způsobem Hamming: $w(n) = 0.54 - 0.46\cos(\pi n / N)$

k – koefient od 0 do $N-1$, kde *k* odpovídá frekvenci $f(k) = kf_s / N$

f_s – vzorkovací frekvence

Spektrální rozsah $|X(k)|$ je v poměru frekvence a rozsahu. Frekvence je rozdělena logaritmicky, použitím Mel filtrovací banky $H(k,m)$:

$$X'(m) = \ln\left(\sum_{k=0}^{N-1} |X(k)| \cdot H(k,m)\right) \tag{11}$$

$$H(k,m) = \begin{cases} 0 & \text{pro} \quad f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} & \text{pro} \quad f_c(m-1) \le f(k) < f_c(m) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)} & \text{pro} \quad f_c(m) \le f(k) < f_c(m+1) \\ 0 & \text{pro} \quad f(k) \ge f_c(m+1) \end{cases} \tag{12}$$

f_c - střední frekvence filtrovací banky, $f_c(m) = 700(10^{\phi_c(m)/2595} - 1)$

ϕ_c - střední frekvence Mel rozsahu, $\phi_c(m) = m \cdot \Delta\phi$

$\Delta\phi = \dfrac{(\phi_{\max} - \phi_{\min})}{(M+1)}$, $\phi_{\max}$ je nejvyšší frekvence Mel rozsahu, $\phi_{\min}$ je nejnižší frekvence Mel rozsahu a *M* je počet filtrovacích bank.

$\phi = 2595\log_{10}\left(\dfrac{f}{700} + 1\right)$, $\phi_{\max}$ se vypočítá pomocí $f_{\max}$ a $\phi_{\min}$ pomocí $f_{\min}$.

Jednotlivé MFCC koeficienty jsou poté vypočítány pomocí diskrétní kosinové transformace (DCT):

$$c(l) = \sum_{m=1}^{M} X'(m)\cos\left(l\frac{\pi}{M}(m - \frac{1}{2})\right) \tag{13}$$

4.3 Parametrizace MP3 souborů

Parametrizace MP3 souborů se provede pomocí koeficientů získaných přímo z MP3 souboru. Jedná se o koeficienty, které jsou uloženy jako výsledný produkt procesu kódování audia do formátu MP3, tzn. MDCT koeficienty zakódované Huffmanovým kódováním. Těchto koeficientů bude v každém vektoru umístěno 104 a budou uloženy jako datový typ `double`. Vzhledem k tomu, že neprovádíme kompletní zpětnou rekonstrukci audio signálu, nebudeme uvažovat překryvání jednotlivých rámců (overlapping), které u formátu MP3 činí 50%.

U všech MP3 souborů byl sjednocen datový tok na 128 kbit/s a byly otestovány jak s použitím jednoho kanálu (mono), tak s použitím dvou kanálů (stereo).

4.3.1 read_mp3.c

Tento program načte MP3 soubor a zjistí následující parametry:

- velikost souboru
- existenci a velikost TAGu ID3 verze 2
- použitý datový tok
- použitý kanálový mód
- velikost rámce
- celkový počet rámců

Na základě těchto parametrů pak program načítá pouze koeficienty „čistých“ audio dat a ukládá je do výstupního souboru, který je použit následujícím programem.

4.3.2 create_htkfea.cpp

Tento program vytvoří soubor, do kterého umístí vektory s koeficienty ze souboru vytvořeného programem *read_mp3* a opatří jej HTK hlavičkou, definující použití uživatelského nastavení (USER).

4.4 Vyhodnocení

Aby bylo možné porovnat navržený způsob rozpoznávání hudebních stylů z MP3 a ověřit funkčnost HTK nástrojů, byly nejprve všechny trénovací a testovací vzorky převedeny do formátu WAV, který HTK nástroje přímo podporují (SOURCEFORMAT = WAV) a parametrizace proběhla standardně pomocí nástroje *HCopy* s výslednými MFCC koeficienty.

Velikost vektoru parametrů byla v tomto případě 31, zvedání počtu komponent pomocí editoru *HHed* probíhalo s krokem 4, s počáteční hodnotou 4. V závislosti na následném přetrénování pomocí *HERest* a optimálním výsledku rozpoznání pomocí *HVite*, se konečná hodnota počtu komponent ustálila na hodnotě 24. Maximální přípustná hodnota zvýšení komponent činila 52, v dalších krocích již byla nástrojem *HHEd* hlášena varování. Provedlo se tedy 6 cyklů zvýšení komponent a přetrénování.

Celková úspěšnost rozpoznávání byla v tomto případě 48%. Nejlépe byly rozpoznány vzorky z kategorie *Hard Rock & Metal*, kde úspěšnost dosáhla 100%. Tento žánr má jednoznačný charakter projevu, proto není příliš obtížné jej rozpoznat. Na pomyslném druhém místě se umístila kategorie *Big Band & Orchestra*. Zde byly správně rozpoznány 4 vzorky z 5. Zajímavé bylo přiřazení 5. vzorku do kategorie *Hard Rock & Metal*. Jednalo se rychlý a hodně „divoký“ Jump Jive, což zřejmě rozpoznávač „zmátlo“, proto jej nezařadil správně.

Nejhůře dopadly kategorie *Relax & Dream* a *Latin*. Vzorky z těchto kategorií byly většinou řazeny do kategorie *8 Beat & Oldies*, *Pop Ballad* a *Country*.

Je velmi důležité zmínit, že některé vzorky by se daly zařadit i do jiné kategorie, než pro které jsou primárně definovány, což ukázalo i rozpoznávání. I když nebyl vzorek správně rozpoznán, tak výsledné zařazení se ve většině případů dalo bez problému akceptovat a celková úspěšnost by tedy mohla být o poměrně značné procento vyšší.

Další etapou bylo rozpoznávání samotných MP3 souborů s parametrizací popsanou v kapitole 5.3. Nejprve bylo nutné upravit hodnoty u každého prototypu modelu:

```
~o <VecSize> 104 <USER>
~h "BEATOLDIES"
<BeginHMM>
<NumStates> 3
<State> 2 <NumMixes> 1
<Mean> 104
0.0 0.0 0.0 ...
<Variance> 104
1.0 1.0 1.0 ...
<TransP> 3
0.0 1.0 0.0
0.0 0.6 0.4
0.0 0.0 0.0
<EndHMM>
```

Pak následoval stejný postup, jako u WAV souborů. Zvedání počtu komponent s krokem 4 nevedlo k příliš příznivým výsledků, proto byl nakonec krok stanoven na hodnotu 1, kde počáteční hodnota byla 2 a konečná hodnota 7. Zde se tedy provedlo 7 cyklů zvýšení komponent a přetrénování. Maximální přípustná hodnota zvýšení komponent činila 23, v dalších krocích již byla nástrojem *HHEd* hlášena varování.

Celková úspešnost rozpoznávání činila 28%. Jednoznačně nejlépe zde dopadly vzorky z kategorie *Hard Rock & Metal*, nerozponaný vzorek z této kategorie byl určen jako *Contemporary Rock & Big Beat*, což je také rockový žánr. Nejhůře dopadly kategorie *Relax & Dream* a *Rock Ballad*, kde úspěšnost činila 0%. Zajímavé bylo zjištění, že označení *Rock Ballad* se nevyskytlo u žádného ze 60 rozpoznávaných vzorků a to téměř ve všech úrovních přetrénování. *Relax & Dream* se vyskytoval u kategorií *Pop Ballad*, *Country* a *Latin*.

Závěrečnou etapou bylo rozpoznávání mono MP3 souborů. Výsledky však dopadly nejhůře ze všech. 5 kategorií nebylo vůbec rozpoznáno a u zbývajicích činila úspěšnost 20%, tedy 1 vzorek z 5. Následující tabulka kompletně shrnuje procento úspěšného rozpoznání ve všech etapách.

	WAV	**MP3**	**MP3**
	1411 kbit/s, stereo	***128 kbit/s, stereo***	***128 kbit/s, mono***
8 Beat & Oldies	40%	20%	20%
Pop Ballad	40%	20%	0%
Relax & Dream	20%	0%	0%
Contemporary Rock & Big Beat	40%	40%	20%
Hard Rock & Metal	100%	80%	20%
Rock Ballad	40%	0%	0%
Disco & Schlager	40%	20%	0%
Trance & Techno	60%	40%	20%
Hip Hop & Rap	40%	20%	0%
Big Band & Orchestra	80%	40%	20%
Country	60%	20%	20%
Latin	20%	40%	20%
CELKOVÁ ÚSPĚŠNOST	48%	28%	12%

Tabulka 2: Procentuální úspěšnost rozpoznávání

5 Závěr

Výsledky ukázaly, že koeficienty z MP3 souborů lze využít pro parametrizaci za účelem rozpoznávání hudebních stylů, aniž by to vyžadovalo složitých dekódovacích procesů.

Je nutné zmínit, že možnou nevýhodou jsou v této fázi poměrně limitované hodnoty těchto koeficientů. Pokud by se tyto koeficienty dále dekódovaly pomocí Huffmanových tabulek, je možné získat mnohem větší rozsah hodnot, jak v kladném směru, tak i záporném, což by v konečném stádiu zvýšilo přesnost rozpoznávání a bylo tak dosaženo mnohem lepších výsledků. Tento dekódovací proces však není triviální záležitost, vyžaduje spoustu času a přesné výpočty.

V této práci jsou použity vzorky MP3 souborů s konstantním datovým tokem. Správné použití MP3 souborů s variabilním datovým tokem vyžaduje další úsilí. Struktura takového souboru není jednoznačně dána, dynamicky se mění a u každého rámce jsou nezbytné další výpočty.

Počet kategorií by sice mohl být vyšší, avšak pokud srovnáme podobně zaměřené práce, kde počet kategorií nepřesáhne 10, je zvolený počet 12 více než dostačující.

Co se týká použitých vzorků, tak naprostá většina z nich mi byla známa, což snížilo časovou náročnost při hledání. Zbývající vzorky byly vybrány náhodně a akceptovány na základě poslechu. Žánrové členění nebyl pro mě jako muzikanta žádný problém.

Pro zvýšení variability by určitě bylo vhodné zvýšit počet vzorků v každé kategorii, např. 50 nebo 100 vzorků by určitě přesněji specifikovalo danou kategorii, jenže s tím by celkově velmi rychle narostla také časová náročnost, ať už se jedná o shánění vzorků, parametrizování nebo rozpoznávání.

Literatura

[1] AUCOUTURIER, J-J.: *Representing Musical Genre: A State of the Art*. Paris: SONY Computer Science Laboratory. 2003. [online], [cit. 2009-05-03].
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.102>

[2] BOUVIGNE, G., *MP3‘ Tech*. 2007. [online], [cit. 2009-05-03].
URL: <http://www.mp3-tech.org>

[3] COX, K.: *A Guide to Open Source Audio Streaming*. [online], [rev. 2008-06-17], [cit. 2009-05-03].
URL: <http://www.gnuware.com/icecast/book1.html>

[4] ČERNOCKÝ, J.: *Zpracování řečových signálů*. [online], [cit. 2009-05-03].
URL: <http://www.fit.vutbr.cz/study/courses/ZRE/public/.cs>

[5] DANNENBERG, R. B.: *A Machine Learning Approach to Musical Style Recognition*. Carnegie Mellon University. 1997. [online], [cit. 2009-05-03].
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.527>

[6] HORČÍK, Z.: *Metody kompenzace šumu pro rozpoznávače řeči využívajících skrytých Markovových modelů*. Praha. [online], [cit. 2009-05-03].
URL: <http://noel.feld.cvut.cz/vyu/tss/hmm9.htm>

[7] LAI, H-CH.: *Real-Time Implementation of MPEG-1 Layer 3 Audio Decoder n a DSP Chip*. Diplomová práce, National Chiao-Tung University, Taiwan. [online], [cit. 2009-05-03].
URL: <http://www.mp3-tech.org/programmer/docs/thesis_lai.pdf>

[8] O’NEILL, D.: *ID3.org*. [online], [rev. 2009-01-13], [cit. 2009-05-03].
URL: <http://www.id3.org>

[9] PAMPALK, E.: *Content-based Organization and Visualization of Music Archives*. Vienna: Austrian Research Institute for Artificial Intelligence. 2002. [online], [cit. 2009-05-03].
URL: <http://www.ifs.tuwien.ac.at/ifs/research/pub_pdf/pam_acmmm02.pdf>

[10] PONCE DE LEÓN, P. J.: *A shallow description framework for musical style recognition*. [online], [cit. 2009-05-03].
URL: <http://grfia.dlsi.ua.es/repositori/grfia/pubs/2/ssspr04.pdf>

[11] POPP, H.: *An introduction to MPEG Layer 3*. Erlangen: Fraunhofer Institute for Integrated Circuits, Germany. 2000. [online], [cit. 2009-05-03].
URL: <http://www.ebu.ch/trev_283-popp.pdf>

[12] RAISSI, R.: *The Theory Behind Mp3*. 2002. [online], [cit. 2009-05-03].
URL: <http://www.mp3-tech.org/programmer/docs/mp3_theory.pdf>

[13] SIGURDSSON, S., PETERSEN, K. B.: *Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music*. Technical University of Denmark, Lyngby. [online], [cit. 2009-05-03].
URL: <http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4690/pdf/imm4690.pdf>

[14] WHITMAN, B., SMARAGDIS, P.: *Combining Musical and Cultural Features for Intellingent Style Detection*. Massachusetts: Cambridge. 1999. [online], [cit. 2009-05-03].
URL: <http://alumni.media.mit.edu/~bwhitman/whitman02combining.pdf>

[15] YOUNG, S.: *The HTK Book*. Cambridge University Engineering Department, Cambridge, 2006. [online], [cit. 2009-05-03].
URL: <http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml>

[16] *Inside the MP3 codec*. Cambridge University Engineering Department, Cambridge, 2006. [online], [cit. 2009-05-03].
URL: <http://www.mp3-converter.com/mp3codec>

[17] *Lame MP3 encoder*. [online], [cit. 2009-05-03].
URL: <http://lame.sourceforge.net>

[18] *Wapedia*. [online], [cit. 2009-05-03].
URL: <http://wapedia.mobi/en/MP3>

[19] *Wikipedia*. [online], [cit. 2009-05-03].
URL: <http://en.wikipedia.org/wiki/MP3>

Seznam příloh

- Příloha A
- Příloha B
- Příloha C
- Příloha D
- Příloha E
- CD

Příloha A

Detailní popis Huffmanových tabulek.

Charakteristika 32 Huffmanových tabulek						
Index tabulky	**Max. hodnota**	**Index tabulky**	**Max. hodnota**	**linbits**	**Max. hodnota**	**Oblast**
A	1	B	1	Ne		count_1
0	0	16	15	1	16	
1	1	17	15	2	19	
2	2	18	15	3	23	
3	2	19	15	4	31	
4	není použita	20	15	6	79	
5	3	21	15	8	271	
6	3	22	15	10	1039	
7	5	23	15	13	8207	
8	5	24	15	4	31	
9	5	25	15	5	47	
10	7	26	15	6	79	
11	7	27	15	7	143	
12	7	28	15	8	271	
13	15	29	15	9	527	
14	není použita	30	15	11	2016	
15	15	31	15	13	8207	big_value

Tabulka A1: Charakteristika 32 Huffmanových tabulek:

Příloha B

Detailní popis záhlaví rámce MP3 souboru.

MPEG version ID	
Hodnota	***Popis***
00	MPEG version 2.5 (není oficiálním standardem)
01	Rezervováno
10	MPEG verze 2
11	MPEG verze 1

Tabulka B1: MPEG version ID

Layer	
Hodnota	***Popis***
00	Rezervováno
01	Layer 3
10	Layer 2
11	Layer 1

Tabulka B2: Layer

CRC Protection	
Hodnota	***Popis***
0	Obsažen kontrolní součet
1	Bez kontrolního součtu

Tabulka B3: CRC Protection

Bitrate index	
Hodnota	***Popis***
0000	Volné
0001	32 kbit/s
0010	40 kbit/s
0011	48 kbit/s
0100	56 kbit/s
0101	64 kbit/s
0110	80 kbit/s
0111	96 kbit/s
1000	112 kbit/s
1001	128 kbit/s
1010	160 kbit/s
1011	192 kbit/s
1100	224 kbit/s
1101	256 kbit/s
1110	320 kbit/s
1111	Špatné

Tabulka B4: Bitrate index

Sampling rate frequency index	
Hodnota	***Popis***
00	44100 Hz
01	48000 Hz
10	32000 Hz
11	Rezervováno

Tabulka B5: Sampling rate frequency index

Channel mode	
Hodnota	***Popis***
00	Stereo
01	Joint Stereo
10	Dual
11	Mono

Tabulka B6: Channel mode

Padding	
Hodnota	***Popis***
0	Rámec bez paddingu
1	Rámec s paddingem

Tabulka B7: Padding

Mode extension *(only if Joint Stereo is set)*		
Hodnota	***Popis***	
	Intensity Stereo	***MS Stereo***
00	Vypnuto	Vypnuto
01	Zapnuto	Vypnuto
10	Vypnuto	Zapnuto
11	Zapnuto	Zapnuto

Tabulka B8: Mode extension

Copyright	
Hodnota	***Popis***
0	Audio nemá autorská práva
1	Audio má autorská práva

Tabulka B9: Copyright

Original	
Hodnota	***Popis***
0	Kopie originálního média
1	Originalní médium

Tabulka B10: Original

M - Emphasis	
Hodnota	***Popis***
00	Nic
01	50/15
10	Rezervováno
11	CCIT J.17

Tabulka B11: Emphasis

Příloha C

Detailní popis struktury postranních informací v záhlaví MP3 souboru.

Skupina	Soubor měřítek
0	0,1,2,3,4,5
1	6,7,8,9,10
2	11,12,13,14,15
3	16,17,18,19,20

Tabulka C1: Scfsi

scalefac_compress	slen1	slen2
0	0	0
1	0	1
2	0	2
3	0	3
4	3	0
5	1	1
6	1	2
7	1	3
8	2	1
9	2	2
10	2	3
11	3	1
12	3	2
13	3	3
14	4	2
15	4	3

Tabulka C2: Scalefac_compress

block_type	typ okna
00	zakázáno
01	začátek
10	3 krátká okna
11	konec

Tabulka C3: Block_type

Příloha D

Detailní popis struktury rámce MP3 souboru s variabilním datovým tokem.

Struktura prvního rámce	
Byte	***Obsah***
0 - 3	Standardní záhlaví rámce. Většinou obsahuje hodnoty FF FB 30 4C, délka rámce je 156 Bytů a to je přesně velikost místa pro uložení informací o variabilním datovém toku. Záhlaví obsahuje důležité informace, platné pro celý soubor: - MPEG (MPEG1 nebo MPEG2) - Index vzorkovací frekvence (Sampling rate frequency index) - Kanál (např. JointStereo)
4 - x	Není využit až do řetězce „Xing“ (58 69 6E 67). Tento řetězec se používá jako hlavní identifikátor souboru s variabilním datovým tokem. Pokud není řetězec nalezen, předpokládá se, že se jedná o soubor s konstatním datovým tokem. Tento řetězec může být umístěn v různých částech, odpovídající hodnotám MPEG a kanálu.
13 - 16	„Xing“ pro MPEG2 a kanál mono.
21 - 24	„Xing“ pro MPEG2 a kanál, který není mono.
21 - 24	„Xing“ pro MPEG1 a kanál mono.
36 - 39	„Xing“ pro MPEG1 a kanál, který není mono. (nejvíce používané)

Tabulka D1: Struktura 1. rámce

Schéma pro MPEG1 a kanál, který není mono		
Byte	***Obsah***	
40 - 43	***Příznaky (Flags)***	
	Hodnota	***Název***
	00 00 00 01	Frames Flag
	00 00 00 02	Bytes Flag
	00 00 00 04	Table of Contents (TOC) Flag
	00 00 00 08	Variable Bit Rate (VBR) Flag

Tabulka D2: Schéma pro MPEG1 a kanál, který není mono

Schéma pro MPEG1 a kanál, který není mono (pokračování)	
Byte	***Obsah***
44 - 47	***Rámce (Frames)***
	Počet rámců v souboru, včetně prvního informačního.
48 - 51	***Byty (Bytes)***
	Velikost souboru v bytech.
52 - 151	***Obsah (TOC - Table of Contents)***
	100 indexů o velikosti jednoho bytu pro snažší vyhledávání v souboru. Každý byte má hodnotu, která odpovídá výrazu: (TOC[i] / 256) * FileLengthInBytes *Př.* Ze skladby uběhlo 240 sekund a skočí se na 60. sekundu. Velikost souboru je 5 000 000 bytů. TOC[(60/240)*100]=TOC[25], odpovídající byte v souboru je tedy (TOC[25]/256)*5000000
152 - 155	***VBR Scale***
	Hodnota udávající kvalitativní hodnotu v procentech. 0% - nejlepší, 100% - nejhorší

Tabulka D3: Schéma pro MPEG1 a kanál, který není mono (pokračování)

Příloha E

Detailní popis struktury TAGů.

Struktura TAGu ID3 verze 1		
Byty	***Délka***	***Obsah***
0 - 2	3	Indentifikátor TAGu, platný TAG musí obsahovat řetězec "TAG"
3 - 32	30	Název skladby
33 -62	30	Interpret
63 - 92	30	Album
93 - 96	4	Rok
97 - 126	30	Komentář
127	1	Žánr

Tabulka E1: Struktura TAGu ID3 verze 1

Záhlaví TAGu ID3 verze 2	
Byty	***Obsah***
0 - 2	Indentifikátor TAGu, platný TAG musí obsahovat řetězec "ID3"
3 - 4	Verze TAGu
5	Příznaky (Flags)
6 - 9	Velikost TAGu

Tabulka E2: Záhlaví TAGu ID3 verze 2

Frame header	
Byty	***Obsah***
0 - 3	Identifikátor rámce (Frame identifier)
4 - 7	Velikost (Size)
8 - 9	Příznaky (Flags)

Tabulka E3: Záhlaví rámce

Frame identifier	
Identifikátor	***Popis***
TRCK	Číslo stopy
TENC	Kodér
WXXX	URL
TCOP	Identifikátor rámce
TOPE	Původní interpret
TCOM	Skladatel
TCON	Žánr
COMM	Komentář
TYER	Rok
TALB	Album
TPE1	Interpret
TIT2	Název skladby

Tabulka E4: Identifikátor rámce

Příznaky (Flags)	
Příznak	***Popis***
A	Ochrana TAGu proti změnám (TAG alter preservation)
B	Ochrana souboru proti změnám (File alter preservation)
C	Pouze pro čtení
I	Komprese
J	Šifrování
K	Skupinová identita

Tabulka E5: Příznaky

Seznam žánrů (Genre list)					
Číslo	***Žánr***	***Číslo***	***Žánr***	***Číslo***	***Žánr***
1	Blues	43	Punk	85	Bebob
2	Classic Rock	44	Space	86	Latin
3	Dance	45	Meditative	87	Revival
4	Disco	46	Instrumental Pop	88	Celtic
5	Funk	47	Instrumental Rock	89	Bluegrass
6	Grunge	48	Ethnic	90	Avantgarde
7	Hip-Hop	49	Gothic	91	Gothic Rock
8	Jazz	50	Darkwave	92	Progressive Rock
9	Metal	51	Techno-Industrial	93	Psychedelic Rock
10	New Age	52	Electronic	94	Symphonic Rock
11	Oldies	53	Pop-Folk	95	Slow Rock
12	Jiný (Other)	54	Eurodance	96	Big Band
13	Pop	55	Dream	97	Chorus
14	R&B	56	Southern Rock	98	Easy Listening
15	Rap	57	Comedy	99	Acoustic
16	Reggae	58	Cult	100	Humour
17	Rock	59	Gangsta	101	Speech
18	Techno	60	Top 40	102	Chanson
19	Industrial	61	Christian Rap	103	Opera
20	Alternative	62	Pop/Funk	104	Chamber Music
21	Ska	63	Jungle	105	Sonata
22	Death Metal	64	Native American	106	Symphony
23	Žertovné (Pranks)	65	Cabaret	107	Booty Bass
24	Soundtrack	66	New Wave	108	Primus
25	Euro-Techno	67	Psychadelic	109	Porn Groove
26	Ambient	68	Rave	110	Satire
27	Trip-Hop	69	Showtunes	111	Slow Jam
28	Vocal	70	Trailer	112	Club
29	Jazz+Funk	71	Lo-Fi	113	Tango
30	Fusion	72	Tribal	114	Samba
31	Trance	73	Acid Punk	115	Folklore
32	Classical	74	Acid Jazz	116	Ballad
33	Instrumental	75	Polka	117	Power Ballad
34	Acid	76	Retro	118	Rhythmic Soul
35	House	77	Musical	119	Freestyle
36	Game	78	Rock & Roll	120	Duet
37	Sound Clip	79	Hard Rock	121	Punk Rock
38	Gospel	80	Folk	122	Drum Solo
39	Noise	81	Folk-Rock	123	A capella
40	AlternRock	82	National Folk	124	Euro-House
41	Bass	83	Swing	125	Dance Hall
42	Soul	84	Rast Fusion		

Tabulka E6: Seznam žánrů

Obsah CD

- `/cfg`: obsahuje konfigurační soubor pro nástroj HCopy
- `/dics`: obsahuje slovník pro nástroj HVite
- `/hed`: obsahuje konfigurační soubor pro nástroj HHEd
- `/hmm0 – /hmm3`: obsahují přetrénované modely
- `/htk`: zkompilovaná sada nástrojů HTK
- `/lists`: obsahuje seznam modelů
- `/mlf`: obsahuje Master-Label soubory
- `/net`: obsahuje síť pro rozpoznávání
- `/proto`: obsahuje prototypy jednotlivých modelů
- `/scripts`: obsahuje skripty pro trénování
- `/USR`: složka pro uživatelské soubory
- `create_htkfea.cpp`: zdrojový kód programu
- `IBP.pdf`: elektronická verze práce
- `Makefile`
- `make_mlf.pl`: skript pro tvorbu Master-Label souborů
- `PLAYLIST.pdf`: seznam použitých MP3 vzorků
- `read_mp3.c`: zdrojový kód programu
- `README`: návod k obsluze
- `recognize.sh`: skript pro spuštění rozpoznávání