



# User:duchon.gabriel

## Vyhľadávanie melódie v MIDI súboroch

### Popis projektu a motivácia

Výstupom tohto projektu je desktopová aplikácia, ktorá indexuje množinu MIDI súborov a vyhľadáva používateľom zadanú melódiu.

MIDI súbory uchovávajú melódiu, presnejšie noty a podľa toho koľko notových stop (Track) obsahuje skladba, môže byť zaznamenaná melódia monofonická (ak je prítomná 1 stopa), alebo polyfonická (ak je prítomných viac stôp). Keďže polyfonické melódie sú rozšírenejšie a aj zložitejšie z pohľadu štruktúry, v mojom projekte sa budem venovať vyhľadávaniu aj polyfonických melódií. Pri prehrávaní polyfonickej melódie sa jednotlivé stopy prehrávajú súčasne (pre každú stopu je možné definovať ľubovoľný hudobný nástroj) a tým vytvárajú hudbu viacerými hudobnými nástrojmi.

Najčastejšia reprezentácia melódie pri vyhľadávaní je sekvencia nôt formou textového reťazca, kde každé „slovo“ reprezentuje jednu notu, presnejšie jej výšku a dĺžku.

Pomocou takejto reprezentácie môžeme ďalej pracovať s melódiou ako s textom, a použiť nástroje na indexovanie a vyhľadávanie textových reťazcov.

Keďže máme viac tokov informácií (oproti jednoduchým textovým zdrojom), jednotlivé stopy sa treba spracovať zvlášť, to znamená, že z jednej polyfonickej MIDI skladby sa vygeneruje viac textových reťazcov. Niektoré stopy (na základe analýzy ich obsahu) sa môžu ignorovať; napr.: stopy neobsahujúce noty (len nastavenia MIDI súboru), stopy v ktorých sú noty bicích nástrojov, alebo noty vytvárajúce zvukové efekty (telefón, pískanie, siréna, atď.), ...

Je mnoho desktopových aplikácií, ktoré podporujú spracovanie MIDI súborov (a tiež hudobné skladby v inom formáte;) a tiež veľa webových portálov, ktoré umožňujú vyhľadávanie melódie. Chýbajú však aplikácie, ktoré by boli desktopové a umožňovali by vyhľadávanie nad bázou skladieb, ktoré nie sú prítomné v báze jednotlivých webových portálov. Jednotlivé online vyhľadávače nepodporujú pridanie do bázy skladieb vlastnú skladbu, tým pádom môžeme vyhľadávať melódiu len nad tou množinou, ktorú máme danú.

Reprezentácia a vyhľadávanie melódie sú podobné k textovej reprezentácii a vyhľadávaniu v textových zdrojoch. Hlavným problémom tejto oblasti je návrh vhodnej reprezentácie melódie, ktorá by obsahovala práve tie informácie, ktoré sú potrebné na identifikáciu melódie. Najčastejšia reprezentácia melódie je formou textu, keďže umožňuje použiť oveľa viac znakov ako číselné vyjadrenie a následne môžeme používať už existujúce a výkonné nástroje na indexovanie a vyhľadávanie informácií.

## Prehľad súčasných riešení daného problému

Existuje pomerne veľa webových aplikácií na vyhľadávanie melódie, niektoré sú väčšie niektoré menšie, ale ani jedna z nich neumožňuje aplikovať vyhľadávanie na vlastnej množine skladieb.

Najznámejší portál na vyhľadávanie melódie je musipedia – The Open Music Encyclopedia. Slúži na vyhľadávanie hudobných skladieb podľa zadanej melódie, vyhľadávanie na základe hudobnej kontúry alebo aj na základe rytmu. Počet melódií a indexovaných skladieb portál neuvádza, ale na základe testovania repertoár skladieb je pestrý keďže na portáli je možné vyhľadať klasické, jazzové, populárne a tiež aj skladby iných žánrov, ale najviac výsledkov bolo klasického žánru. Umožňuje rôzne formy zadávania dopytu (virtuálna klávesnica, notový zápis, Parsonov kód, contour search, hmkanie z mikrofónu, vyhľadávanie na základe rytmu). Na vyhľadávanie využíva nástroj Melody hound, ktorý ale detailnejšie neopisuje.

## Textová reprezentácia melódie a Parsonov kód

Pri transformácii melódie do textového reťazca sa použijú nasledovné faktory tónov:

- výška tónu (väčšinou sa reprezentuje ako písmeno abecedy) / pauza
- dĺžka tónu (číselná reprezentácia)
- oddeľovač (väčšinou medzera)

Pri niektorých riešeniach sa prvý, alebo druhý faktor ignoruje, z dôvodu zjednodušenia zadávania dotazu na vyhľadávanie používateľom, ktorí sú hudobne menej zdatný (napr. dĺžka tónu často nie je dodržaná). Zaujímavé je, že napriek vynechaniu jednej z faktorov sú tieto vyhľadávania tiež efektívne, čo môžeme spozorovať v praxi (portál musipedia tiež podporuje vyhľadávanie na základe melódie aj rytmiky). Podobný princíp využíva aj metóda vyhľadávania melódie pomocou Parsonovho kódu. Reprezentácia skladby a dopyt vyhľadávania touto formou neuchováva dĺžku tónov, ani hodnotu výšky tónov. Jediný typ údajov v tejto reprezentácii je smer zmeny výšky tónu medzi jednotlivými susednými notami, ktorá sa eviduje nasledovne:

- znak \* sa použije ako označenie prvého tónu
- znak D (down) sa používa na označenie nôt, ktoré majú nižšiu výšku, ako predchádzajúca nota
- znak U (up) sa používa na označenie nôt, ktoré majú vyššiu výšku, ako predchádzajúca nota
- znak R (repeat) sa používa na označenie nôt, ktoré majú rovnakú výšku, ako predchádzajúca nota

## Vyhľadávanie melódie použitím n-gramov

N-gram je pojem používaný v oblasti vyhľadávania informácií. Pojmom n-gram sa označuje reťazec získaný z n počet susedných znakov v texte. V oblasti hudby je možné použiť tento prístup na získanie menších častí z textovej sekvencie, ktorá reprezentuje melódiu. Pomocou n-gramov môžeme vyhľadávať informácie efektívnejšie, keďže sa

navzájom porovnávajú menšie bloky (menej náročný proces na pamäť). Tieto bloky sú výstupom klzavého okna, veľkosti  $n$ , ktorá extrahuje jednotlivé bloky dát tiež veľkosti  $n$ .

## Popis riešenia, použitý softvér

Pri indexovaní a vyhľadávaní sa noty prevedú do alfanumerickej podoby, z dôvodu, aby sme mohli využiť existujúce efektívne nástroje na vyhľadávanie. Do alfanumerickej reprezentácie neprevedieme všetky údaje nôt z MIDI súboru, keďže niektoré údaje by boli pre vyhľadávanie zbytočné, preto zohľadníme len dva parametre nôt: výšku tónu a trvanie tónu. Výška tónu nebude vyjadrená jej konkrétnou hodnotou, ale ako rozdiel/zmena výšky tónu aktuálnej a predchádzajúcej noty. Pomocou evidovania zmeny výšky nôt budeme môcť vyhľadávať melódie v rôznych transpozíciách skladby (pozn.: ak zmeníme výšku tónu každej noty o danú hodnotu tak skladba bude obsahovať vyššie tóny, ale melódia sa pre pozorovateľa nezmení, preto melódia je nezávislá od výšky jednotlivých nôt, ale len od zmien susedných nôt).

Alfanumerická reprezentácia jednej noty bude nasledovná:

sign + deltaV + delay

sign: P - ak sa výška tónu zmenila kladnou hodnotou

N - ak sa výška tónu zmenila zápornou hodnotou

O - ak sa výška tónu nezmenila

deltaV: absolútna hodnota zmeny výšky tónu (dvojciferná hodnota max 99)

delay: trvanie tónu (dvojciferná hodnota max 99)

Na indexovanie a vyhľadávanie budem používať knižnicu Apache Lucene 3.4.0. Počas predmetu Diplomový projekt 1 som sa oboznámil (aj) s literatúrou [1,2,3] v ktorej bol riešený princíp vyhľadávania melódie pomocou  $n$ -gramov. Na tokenizáciu alfanumerickej reprezentácie hudobnej stopy budem používať triedu NGramTokenizer, ktorý rozdelí textový reťazec na množinu  $n$ -gramov. Pri vyhľadávaní sa z vyhľadávanej melódie taktiež vygenerujú  $n$ -gramy, ktoré sa následne vyhľadajú.

## Vytvorenie textovej reprezentácie melódie

Pri načítaní MIDI súboru sa vygeneruje vlastná objektová štruktúra, ktorá nám umožňuje efektívnejšie získať jednotlivé informácie nôt, oproti MIDI reprezentácii, kde sú jednotlivé informácie nôt vyjadrené zvlášť ako udalosti (MidiEvent) a pre ich získanie je nutné prehľadať celý zoznam udalostí. Po vytvorení vlastnej reprezentácie máme rýchly prístup ku každej MIDI stope a v rámci nich ku každej note a k jej vlastnostiam, ktorú použijeme okrem iných aj pri generovaní textových reprezentácií stôp.

Textová reprezentácia melódie sa generuje vždy z jednej MIDI stopy, nie z viacerých, a z jednej MIDI stopy sa môžu vygenerovať aj viaceré textové reťazce (maximálne 4). Ich počet závisí od toho, aký je maximálny počet súčasne prehrávaných nôt v stope.

Výstupom tohto algoritmu sú textové reťazce obsahujúce alfanumerické reprezentácie jednotlivých nôt oddelené medzerou.

## Implementácia

Projekt som podľa plánov implementoval v Jave a použil som knižnicu `lucene-core-3.4.0.jar` a ďalšie jej doplnujúce knižnice: `lucene-analyzers-3.4.0.jar` a `lucene-queryparser-3.4.0.jar`

Doplnujúce knižnice som využil len pri implementovaní funkcií na vyhľadávanie dopytu. Ako prvé som použil triedu `NGramTokenizer` z knižnice `lucene-analyzers-3.4.0.jar`, ktorou som vytváral n-gramy rôznej dĺžky z vyhľadávaného reťazca. Po čase sa ukázalo, že by bolo efektívnejšie vytvárať n-gramy, v ktorých budú uvedené celé reprezentácie nôt a nie aj odseknuté časti, z dôvodu, že neúplné reprezentácie nôt môžu zavádzať pri vyhľadávaní.

Z tohto dôvodu som zmenil stratégiu indexovania a vyhľadávania na nasledovnú [4]: indexovať sa budú celé textové reťazce reprezentujúce jednu sekvenciu melódie (jedna celá MIDI stopy, alebo časť jednej MIDI stopy) a až pri vyhľadávaní budeme vytvárať n-gramy presnejšie n-gramy tokenov (shingle) z dopytovanej melódie. Na tento účel som použil triedu `ShingleMatrixFilter` (knižnica `lucene-analyzers-3.4.0.jar`). Z jednotlivých n-gramov tokenov sa vytvoria inštancie triedy `ComplexPhraseQueryParser` (knižnica `lucene-queryparser-3.4.0.jar`), ktorá umožňuje používať náhradné znaky (wildcards) aj vo frázach, čo pri štandardných `QueryParser`-och nie je umožnené (viď. [Lucene Query Parser Syntax](#)).

Výsledky sa zobrazia na konci po uskutočnení vyhľadávani všetkých n-gramov tokenov. Jednotlivé výsledky sa zoradia podľa toho ako často figurovali ako výsledky vyhľadávania n-gramov tokenov.

## Implementácia Nutch pluginu

Pre vytvorenie pluginu pre nástroj Apache Nutch bolo treba implementovať dve triedy. Prvú triedu som nazval *MidiIndexingFilter* a implementuje rozhranie *IndexingFilter* nástroja Nutch, presnejšie jej funkcie *filter*, *getConfig* a *setConf*. Funkcie *getConfig* a *setConf* nastavujú len členskú premennú *conf*, z ktorej sa načítavajú konfiguračné parametre vo funkcii *filter*. Funkcia *filter* obsahuje celý proces konverzie MIDI súboru na textové reťazce a následne ich indexovanie. Druhá trieda, ktorá je súčasťou Nutch pluginov je trieda implementujúca rozhranie *ScoringFilter* v našom prípade názvom *MidiScoringFilter*. Ako som zistil, funkcie tejto triedy sa najčastejšie implementujú spôsobom aby vracali predvolené hodnoty, a taktiež z dôvodu že táto časť nástroja Nutch je slabo dokumentovaná, som sa rozhodol implementovať tieto funkcie podobne tak, aby vracali defaultné hodnoty.

## Popis dát na ktorých bolo riešené testovanie

Riešenie bolo testované na 270 MIDI súboroch (celkovo 572 KB, binárne súbory). Z nich sa vygenerovalo 551 KB textu, ktoré reprezentujú melódie MIDI skladieb.

-Zdrojové skladby boli zo zborového spevníka Jednotný katolícky spevník

-Skladby mali v priemere 1,09 stôp

-Z MIDI súborov sa vygenerovalo 1039 textových reprezentácií melódií (tzn. z jednej melódie sa vygenerovalo v priemere 3,8 textových reprezentácií)

## Vyhodnotenie (slovné subjektívne na nejakých konkrétnych príkladoch)

Keďže široký materiál a prítomnosť melódie je subjektívny, tak by som opísal presnosť a pokrytie len vlastnými slovami a tiež aj argumenty môjho odhadu.

**Presnosť** – vyhľadávanie melódie je porovnateľné s vyhľadávaním presných slov (nevystupuje žiadna modifikácia, synonymá alebo rôzne tvary jedného slova) v presnom poradí, čiže ak zadáme vyhľadávanie presnej frázy, tak vyhľadávač zaručí presnosť výsledkov.

**Pokrytie** – keďže pri vyhľadávaní aplikujeme aj vyhľadávanie pomocou n-gramov, zoznam výsledkov zahŕňa aj skladby ktoré neobsahujú celý vyhľadávaný dopyt, ale len nejakú časť dopytu. Treba však dodať že takéto výsledky majú nižší scoring ako výsledky, v ktorých sa vyskytol aj celý vyhľadávaný dopyt.

## Testovanie

Podľa testovania jedno vyhľadávanie trvá v priemere 36,6 milisekúnd. Ak sa daná melódia rozdelí na viac shingleov, tak je treba tento čas sčítať s počtom vygenerovaných shingleov. Tým pádom pre 15 notový dopyt, pri ktorom sa vygeneruje 14 slovná textová reprezentácia, z ktorého sa vygeneruje 9 (6-gramov) + 8 (7-gramov) + 7 (8-gramov) + 6 (9-gramov) + 5 (10-gramov) + 4 (11-gramov) + 3 (12-gramov) + 2 (13-gramov) + 1 (celý dopyt) = 45 vyhľadávaní =  $45 * 36,6 = 1647$  milisekúnd

Vygenerovanie n-gramov (shingleov) z dopytu trvá približne 50 milisekúnd.

Indexovanie 270 MIDI súborov, spoločne o veľkosti 572 KB trvalo 238 sekúnd (zborové piesne). Vygenerovalo sa 551 KB textu.

Pri druhom teste bolo riešené indexovanie 254 MIDI súborov, spoločne o veľkosti 5,39 MB trvalo 789 sekúnd (hlavne pop/rock piesne). Priemerné vyhľadávanie v tejto druhej báze trvalo 84,2 milisekúnd, pričom už obsahovala oveľa viac indexovaného textu (15,2 MB).

## Testovanie vyhľadávania

Stlačené klávesy: K, K, H, H, G, G, D,

Noty: D5, D5, C5, C5, B4, B4, A4,

Hodnoty nôt: 62, 62, 60, 60, 59, 59, 57,

Textová reprezentácia: O00?? N02?? O00?? N01?? O00?? N02??

Dopyt: "O00?? N02?? O00?? N01?? O00?? N02?? "~1

Výsledky vyhľadávania:

D:\Gabi\Dropbox\Diplomovka\workspace\_diplomovka\Diplomovka\samplemidi\jks\JKS\_075.mid  
O0004 N0104 O0004 N0204 O0004 N0204 O0004 N0104 O0004 N0204 P0304 N0504  
N0508 P0504 N0808 P1804 O0004 N0104 O0004 N0204 O0004 N0204 O0004 N0104  
O0004 N0204 P0304 N0504 N0508 P0504 P0208 O0004 N0204 O0004 P0504 O0004  
N0104 O0004 N0204 O0004 N0204 O0004 P0504 O0004 N0104 O0004 N0204 N0204  
P0204 P0204 P0104 N0508 P1004 N1508 P1004 N0104 N0204 N0204

D:\Gabi\Dropbox\Diplomovka\workspace\_diplomovka\Diplomovka\samplemidi\jks\JKS\_048.mid  
O0004 N0104 O0004 N0204 O0004 N0204 N0208 O0004 N0104 O0004 N0204 O0004  
N0204 P1208 O0004 N0104 O0004 N0204 O0004 N0204 N0208 O0004 N0104 O0004  
N1904 N0204 P1904 O0004 N0904 P0701 P0408 O0004 N0204 O0004 P0504 O0004  
N0204 N0608 P0504 O0004 N0204 O0004 N0504 P1001 O0004 N0204 N0608 N0404  
P0904 P0304 P0504 N0104 N0204 O0004 N0204 N0708 P0504 N0608 P0504 N0508  
N1604 P1904 O0004 N0204

D:\Gabi\Dropbox\Diplomovka\workspace\_diplomovka\Diplomovka\samplemidi\jks\JKS\_107.mid  
O0004 O0004 O0004 N0208 P0204 P0208 P0104 N0512 N0804 P1301 O0004 P0204  
N1508 N0201 P1701 P0204 N1704 P1501 N0204 P0204 O0012 N0208 O0008 N0108  
O0008 N0208 O0008 P0508 O0008 P0208 N1708 P1504 N0208 N0108 N0717 N0204  
P0704

D:\Gabi\Dropbox\Diplomovka\workspace\_diplomovka\Diplomovka\samplemidi\jks\JKS\_066.mid  
P0408 N2408 P2204 O0004 P0304 N0108 N0404 P0204 N0304 P0104 N0317 N0204  
N0204 N0804 P1508 N1208 P1604 N0208 O0004 P0304 N0108 N2404 P2204 N0304  
P0104 N0717 N0104 P1508 O0004 N0204 O0004 N0104 O0004 N0204 O0004 P0504  
O0004 N0204 O0004 N0104 O0004 N2304 P2101 O0004 N0904 P0701 N1008 P1404  
N0608 P0404 O0004 P0304 N0808 P0704 N0404 N1304 P1501 N0304 P0104 N1017  
P0204

D:\Gabi\Dropbox\Diplomovka\workspace\_diplomovka\Diplomovka\samplemidi\jks\JKS\_043.mid  
P0217 N0704 P0901 P0204 P0104 O0004 O0004 N0308 P0204 P0104 P0204 N0708  
P0508 N0717 N0104 P0304 P0504 O0004 N0204 O0004 N0104 O0004 N0204 N2008  
P2204 O0004 N0204 N0308 N0504 P0102 N0317 N0201 P0201 P1208 N0204 P0204  
P0104 P0204 N0808 P0504 O0008 N0204 P0204 P0104 P0204 N0808 P0504 N0408  
P0904 N0204 N0208 N0104 N0204 N0208 P0417 N0204 P0204 P0104 P0204 N0808  
P0504 O0008 N0204 P0204 P0104 P0204 N0808 P0504 N0408 P0904 N0204 N0208  
N0104 N0204 N0208

## Spustenie, inštalácia softvéru, použitie softvéru

Aplikácia sa spúšťa štandardne, spustením JAR súboru (dvojklikom alebo z konzoly: javaw -jar midiseacher.jar). Hlavná obrazovka obsahuje tlačidlá:

**Open file** - umožní vybrať si jeden MIDI súbor, ktorý následne spracuje, indexuje (ak ešte nebol indexovaný) a umožní prehrať (a následne analyzovať, ale to je už požiadavka z pohľadu Diplomového projektu)

**Open folder** - umožní vybrať si jeden priečinok a následne spracuje a indexuje všetky (doteraz indexované) MIDI súbory v označenom priečinku.

**Start** - prehrá otvorený MIDI súbor

**Pause** - pozastaví prehrávaný MIDI súbor

**Stop** - zastaví prehrávaný MIDI súbor

**Find melody** - spúšťa virtuálnu klávesnicu, pomocou ktorého je možné nahráť melódiu, prehrať nahranú melódiu a následne ju vyhľadať. Po vyhľadávaní sa zobrazia výsledky a tiež ich scoring v prípade ak vyhľadávanie bolo riešené pomocou n-gramov (ak dopytujeme viac ako 6 nôt)

**Manual search** - spúšťa dialógové okno na zadávanie textového dopytu (Parsonov kód alebo textová reprezentácia melódie). Po spustení vyhľadávania sa zobrazia výsledky a tiež ich scoring v prípade ak vyhľadávanie bolo riešené pomocou n-gramov (ak dopytujeme viac ako 6 nôt)

**Show base** - zobrazí tabuľku indexovaných skladieb

Počas používania aplikácia tiež vypisuje rôzne výpisy na konzolu najmä logovacieho charakteru.

## Literatúra

- [1] Doraisamy, S., A Polyphonic Music Retrieval System Using N-Grams, 2004
- [2] Doraisamy, S., Ruger, S. Robust Polyphonic Music Retrieval with N-grams, 2003
- [3] Isikhan, C., Alpkocak, A, Ozer, Y. Polyphonic Music Retrieval: The N-gram Approach, 2004
- [4] Suyoto, I., Uitdenbogerd, A., Simple Efficient N-Gram Indexing For Effective Melody Retrieval, 2005

## Príklady dát:

[sample01.mid](#) - MIDI súbor

[sample01.txt](#) - zoznam nôt v trackoch a ich hlavne informácie (výška, štart, koniec, trvanie)

[sample01\\_notation.png](#) - notový zápis

[sample01\\_pianoroll.png](#) - grafická reprezentácia nôt (os X = čas ; os Y = výška nôt)

[mjackson.mid](#)

[mjackson.txt](#)

[mjackson\\_track1\\_pianoroll.png](#)

[mjackson\\_track2\\_pianoroll.png](#)

[mjackson\\_track3\\_pianoroll.png](#)