Report Letný Semester

Priebeh práce počas semestra

- Oboznámil som sa s fungovaním a dekódovaním mp3 súborov. Naprogramoval som jednoduchý skript na dekódovanie mp3 hlavičiek. Skript vie, okrem iného, z metadát hlavičky vypočítať celkovú dĺžku mp3 frame-u.
- Implementoval som dekodér, ktorý vie funkcie hlavičkového skriptu použiť na dekódovanie a prečítanie celého mp3 súboru. Vytvoril som pomocné štruktúry na jednoduchšiu prácu s metadátami mp3 súboru.
- Pridal som hlavnú funkciu, ktorá rozdelí súbor na rovnaké časti v náhodnom poradí. Jednoduchý brute-force algoritmus spája časti na základe očakávanej dĺžky chýbajúcich dát v každej rozdelenej časti.
- Algoritmus som upravill tak, aby kontroloval správne dĺžky frame-ov idúcich za sebou. Zmenil som hlavnú stratégiu algoritmu tak, aby namiesto spájania možných dvojíc išiel od prvého frame-u (ktorý je známy).
- Pomocou umelej inteligencie som implementoval nový efektívnejší algoritmus na princípe rozdelenia chunku na 2 časti:
  tail – zvyšok neznámeho frame-u, ktorého hlavička nám chýba,
  head – začiatok nedokončeného frame-u, ktorého hlavička (alebo jej časť) je obsiahnutá v chunku.
  Vypočítaním dĺžok týchto častí a pomocou predpokladanej dĺžky frame-u vieme pre väčšinu prípadov jednoducho povedať, či dva chunky môžu byť spojené. Algoritmus buduje graf nad možnými správnymi spojeniami chunkov. Backtrack skúša najprv cesty s najmenším počtom vetiev.
- Opravil som elementárne chyby, ktoré spôsobovali únik pamäte pri behu programu, zdokumentoval som nové funkcie implementované umelou inteligenciou.
- S pomocou umelej inteligencie som implementoval porovnávač mp3 súborov, ktorý kontroluje poradie pôvodného a skonštruovaného súboru.
- Otestoval som algoritmus na viacerých súborov s rôznymi charakteristikami, dĺžkami a bitrate-om.

Testovanie

Najoptimálnejší algoritmus som testoval na viacerých testovacích súboroch. Každý súbor bol rozdelený do rovnako veľkých blokov a náhodne premiešaný, známy bol len prvý blok. Dĺžka jedného bloku bola pri všetkých testoch 1024 B.

Všetky testovacie súbory boli konvertované do konštantného bitrate-u kvôli vysokej chybovosti algoritmu pri riešení súborov s variabilným bitrate-om. Časové obmedzenie pre každé spustenie bolo 30 sekúnd.

Charakteristika testov

- Test1.mp3 – úvod rockovej skladby – obsahuje opakujúce sa časti
    - Dĺžka: 5 sekúnd
    - Veľkosť pôvodného súboru: 70.4 kB
- Test2.mp3 – šum ktorý naberá na hlasitosti – neobsahuje žiadne totožné dáta
    - Dĺžka: 5 sekúnd
    - Veľkosť pôvodného súboru: 128.7 kB
- Test3.mp3 – vysoko skreslený akord elektrickej gitary – základné overenie funkčnosti
    - Dĺžka: 1 sekunda
    - Veľkosť pôvodného súboru: 10.7 kB
- Test4.mp3 – pesnička pre deti pri umývaní zubov
    - Dĺžka: 22 sekúnd
    - Veľkosť pôvodného súboru: 332.3 kB
- Test5.mp3 – jednoduchá melódia na klavíri
    - Dĺžka: 1 minúta 10 sekúnd
    - Veľkosť pôvodného súboru: 1.7 MB
- Test6.mp3 – živé vystúpenie známeho speváka
    - Dĺžka: 3 minúty 12 sekúnd
    - Veľkosť pôvodného súboru: 2.5 MB

Výsledky testovania

	Bitrate (bit/s)	# of Frames	Presence	Position	Correct Runs
test1.mp3	8k	21	93.00%	20.50%	93.00%
test1.mp3	32k	21	94.00%	34.00%	94.00%
test1.mp3	64k	41	85.00%	16.00%	85.00%
test1.mp3	128k	82	Time limit	Time limit	Time limit
test2.mp3	8k	22	Fail	Fail	Fail
test2.mp3	32k	22	Fail	Fail	Fail
test2.mp3	64k	42	85.90%	20.00%	85.90%
test2.mp3	128k	84	72.20%	12.20%	62.40%
test3.mp3	8k	5	100.00%	100.00%	100.00%
test3.mp3	32k	5	100.00%	100.00%	100.00%
test3.mp3	64k	9	100.00%	100.00%	100.00%
test3.mp3	128k	18	Fail	Fail	Fail
test4.mp3	8k	90	92.40%	7.40%	92.40%
test4.mp3	32k	90	92.20%	7.40%	92.20%
test4.mp3	64k	179	82.70%	7.60%	82.60%
test4.mp3	128k	357	Time limit	Time limit	Time limit
test5.mp3	8k	277	92.00%	2.30%	92.00%
test5.mp3	32k	277	92.00%	1.60%	92.00%
test5.mp3	64k	544	81.90%	1.80%	81.90%
test5.mp3	128k	1108	Time limit	Time limit	Time limit
test6.mp3	8k	753	91.90%	0.50%	91.90%
test6.mp3	32k	753	91.90%	0.50%	91.90%
test6.mp3	64k	1506	81.70%	0.50%	81.70%
test6.mp3	128k	3012	Time limit	Time limit	Time limit

# of Frames – Počet nájdených frame-ov vo vstupnom súbore

Presence – Podiel frame-ov pôvodného súboru, ktoré sú obsiahnuté aj niekde vo výstupnom súbore

Position – Podiel frame-ov, ktoré sú na absolútne správnej pozícii

Correct Runs – Podiel frame-ov, ktoré sú spolu s ďalšími susednými frame-ami v správnom poradí, ale nie sú na správnom mieste

Pozorovanie z testov

- Pre súbory s väčším bitrate-om prichádza na hranice časovej zložitosti – pri bitrate 128 kbit/s je to cca 1000 frame-ov za minútu.
- Aj pri náročnejších a dlhších súboroch sa celkom úspešne tvoria kratšie behy frame-ov, ktoré sú navzájom v správnom poradí
- Všetky výstupné súbory mali veľkú väčšinu správnych behov dĺžky 9 alebo 19 – môže naznačovať logickú chybu alebo kolíziu dĺžky frame-ov a dĺžky chunkov.

- Pri konštantnom bitrate je pravdepodobnosť nesprávneho určenia hranice frame-ov veľmi malá – chyby sa prejavili až pri dlhších súboroch - 1 až 3 nesprávne určené hranice.

Možné zlepšenia

- Kvôli problémom s čítaním hlavičiek a rátaním dĺžky frame-ov je algoritmus nespoľahlivý pre mp3 súbory s variabilným bitrate-om.
- Pre väčšie súbory s konštantným bitrate-om je heuristika hlavičiek nepostačujúca na nájdenie riešenia v realistickom čase.
- Pre súbory s konštantným bitrate-om je vysoká pravdepodobnosť nesprávneho určenia head offsetu mp3 frame-u. Problém je vo fungovaní algoritmu – priorizuje dĺžku behov (za sebou správne idúcich spojení), čo nemusí byť vždy správna možnosť. Riešenie je buď zmena politiky funkcie computeChunkMeta, alebo použitie iných dát v chunkoch na zníženie šance nesprávnych určení.
- Algoritmus nevie identifikovať neštandardné a neznáme typy frame-ov v novších mp3 súboroch, ktoré znižujú pravdepodobnosť úspešnosti algoritmu.
- Spracovanie samotných dát v mp3 súboroch by prinieslo značné zlepšenie algoritmu.