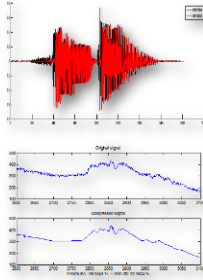


Sažimanje zvučnih datoteka

Prijenos i pohrana podataka imaju svoje trajanje i cijenu. Što je veća količina informacija duže je vrijeme, a veća je i cijena te se zbog toga najveći broj digitalnih podataka pohranjuje i prenosi u najkompaktnijem obliku. Sažimanje podataka je općeniti izraz za različite algoritme i programe koji obrađuju ovaj problem.

1. UVOĐ
- 1.1. Klasifikacija i oblikovanje signala i CGP
- 1.2. Ako živa u većim kanalima i funkcioniraju poput žive
- 1.3. DSD i DSD+ i DSD++
- 1.4. Prilagodba signala za digitalni bitni
2. Zvučni signali i njihove karakteristike (DSD) i oblici
3. Digitalni obrada signala
4. **Sažimanje zvučnih datoteka**
5. Izrazi u obradi zvučnih signala i njihovim izvedenicama
6. Primjena i obrada zvučnih signala i njihovih izvedenica
7. Sažimanje zvučnih datoteka
8. Sažimanje zvučnih datoteka
9. Sažimanje zvučnih datoteka
10. Sažimanje zvučnih datoteka



Sažimanje zvučnih datoteka

Prijenos i pohrana podatak imaju svoje trajanje i cijenu. Što je veća količina informacija duže je vrijeme, a veća je i cijena te se zbog toga najveći broj digitalnih podataka pohranjuje i prenosi u najkompaktnijem obliku. Sažimanje podataka je općeniti izraz za različite algoritme i programe koji obrađuju ovaj problem.

Program za sažimanje koristi se za pretvorbu podataka iz nekog formata, koji je *lagan za uporabu*, u oblik koji je *kompaktan za prijenos i pohranu*.

Program za dekompresiju vraća informacije u izvorni oblik. Najraširenije je šest tehnika za sažimanje podataka. Prve tri tehnike su jednostavne tehnike kodiranja, četvrta je složenija:

- **run-length**,
- **Huffman**,
- **delta kodiranje**
- **LZW (Lempel, Ziv, Welch)**

Posljednje dvije predstavljaju izuzetno složene postupke koje su industrijske norme:

- **JPEG (Joint Picture Expert Group)**
- **MPEG (Motion Picture Expert Group)**

Ovdje nećemo analizirati sažimanje slika i video zapisa jer je to područje drugog kolegija.

Strategije sažimanja podataka

Tablica pokazuje dva različita načina kategoriziranja algoritama sažimanja. U **a) skupini**, postupci su klasificirani kao **bez gubitaka**, ili **s gubicima (lossless ili lossy)**. Tehnike **bez gubitaka** omogućavaju da su podaci nakon restauriranja budu identični izvornim podacima. To je nužna potreba za mnoge tipove podataka, npr.: tablice izvršnog koda, tekstualne datoteke, tablice numeričkih podataka, itd. Ovakvi tipovi informacija ne dozvoljavaju gubitak niti jednog bita. Za usporedbu, datoteke koje predstavljaju slike ili pribavljene signale ne treba držati u idealnim uvjetima za pohranu ili prijenos.

Svi rezultati mjerenja ili akvizicije podataka inherentno sadrže neki iznos šuma. Ako nastale promjene, na ovakvim tipovima signala, nalikuju na mali iznos dodanog šuma ne nastaje velika šteta. Tehnike sažimanja koje dopuštaju ovakvo slabljenje nazivaju se **postupci s gubicima**.

Ova distinkcija je važna jer postupci s gubicima dozvoljavaju mnogo učinkovitije sažimanje od onih bez gubitaka. Što je veći **omjer sažimanja**, podacima se dodaje više šuma.

Drugi način klasifikacije postupaka sažimanja podataka pokazuje **tablica b)**. Najveći broj postupaka radi na način da uzima grupu podataka iz izvorne datoteke, sažima je na neki način, pa zapiše sažete podatke u izlaznu datoteku.

Npr. jedna od tehnika u tablici je **CS&Q, (coarser sampling and/or quantization)** što je kratica za **grublje uzorkovanje i/ili kvantizaciju**. Pretpostavimo da sažimamo digitalizirani valni oblik zvučnog signala kvantiziran 12 bitno. Možemo očitati dva susjedna uzorka iz izvorne datoteke (24 bitne), odbaciti jedan uzorak, odbaciti 4 najmanje značajna bita uzorka pa pohraniti preostalih 7 bitova u izlaznu datoteku. Uz 24 bita na ulazu, a 8 bita na izlazu primijenili smo sažimanje 3:1 korištenjem algoritma s gubicima. Iako je riječ o sirovom postupku, riječ je o vrlo učinkovitom postupku koji se koristi kao temelj JPEG transformacije.

Tablica b) pokazuje CS&Q kao nepromjenjivu ulazno/izlaznu shemu. Dakle, određeni broj bita ulazi se iz ulazne datoteke, a manji broj bitova se zapiše u izlaznu datoteku. Drugi postupci sažimanja dopuštaju čitanje ili pisanje promjenjivog broja bita. Kako obradimo pojedini postupak sažimanja, pogledat ćemo ovu tablicu da vidimo kako se postupak uklapa u klasifikacijsku shemu.

JPEG i MPEG nisu prikazani u tablici jer je riječ o mnogo složenijim algoritmima koji kombiniraju više različitih tehnika te se ne mogu klasificirati u jednostavne kategorije. To je razlog zbog kojeg zahtijevaju odvojenu analizu.

BEZ GUBITAKA	S GUBICIMA	POSTUPAK	VELIČINA GRUPE	
			ulaz	izlaz
Run-length	CS&Q*	CS&Q*	nepromjenjiva	nepromjenjiva
Huffman	JPEG	Huffman	nepromjenjiva	promjenjiva
delta	MPEG	Aritmetički	promjenjiva	promjenjiva
LZW		Run-length, LZW	promjenjiva	nepromjenjiva
a) Bez gubitaka ili s gubicima		b) Nepromjenjiva ili promjenjiva veličina grupe		

Klasifikacija postupaka sažimanja. Postupci sažimanja podataka mogu se podijeliti u dvije grupe. U grupi (a), tehnike se klasificiraju kao bez gubitaka. Postupci bez gubitaka točno restauriraju sažete podatke u obliku identičan izvornom. Postupci s gubicima generiraju aproksimaciju signala. U (b), postupci se klasificiraju prema fiksnoj ili promjenljivoj veličini grupe izvorne datoteke i zapisane u sažetu datoteku.

Run-Length kodiranje

Datoteke podataka najčešće sadržavaju neki znak koji se u retku mnogo puta ponavlja. Tako tekstualne datoteke koriste višestruke praznine između znakova da bi odijelili riječi, rečenice, poglavlja, grafike itd. Digitalizirani signali imaju dijelove iste vrijednosti koje pokazuju da se signal ne mijenja. Digitalizirana glazba ima dugi niz nula u pauzama i među skladovima. Run-length kodiranje je vrlo jednostavni postupak sažimanja takvih tipova datoteka.

Slika pokazuje run-length kodiranje podatkovnog niza sa čestim nulama. Svaki put kad algoritam susretne nulu u ulaznim podacima, zapiše se dvije vrijednosti u izlazne podatke. Prva vrijednost je nula, to je zasticava koja pokazuje početak run-length sažimanja. Druga vrijednost je broj nula u nizu. Ako je niz nula duži od dva izvršit će se sažimanje. U suprotnom slučaju pojedinačne nule bi napravile kodiranu datoteku veću od izvorne.

IZVORNI NIZ PODATAKA	17	8	54	0	0	97	5	16	0	45	23	0	0	0	0	3	67	0	0	8	...
KODIRAN RUN-LENGTH NIZ	17	8	54	0	3	97	5	16	0	1	45	23	0	3	3	67	0	2	8	...	

Primjer run-length kodiranja. Svaki niz nula zamjenjuje se sa dva znaka u sažetoj datoteci: nula koja pokazuje da je riječ o pajavi sažimanja i broja koji pokazuje broj nula u nizu.

Huffmanovo kodiranje

Postupak je imenovan po *Huffmanu* koji je razvio postupak 1950. Slika pokazuje histogram bajt vrijednosti velike ASCII datoteke. Više od 96% ove datoteke sastoji se od samo 31 znaka: mala slova, razmak, zarez, točka. Koristi čemo ovo zapažanje kako bi odredili shemu sažimanja za ovu datoteku. Za početak čemo pridijeliti svakom od 31 znaka pet bitni binarni kod: 00000 = "a", 00001 = "b", 00010 = "c", itd. To nam omogućava da 96% datoteke smanjimo za 5/8. Posljednji pet bitni kod 11111 bit će zastavica koja će ukazivati da je znak koji se prenosi različit od uobičajena 31 znaka. Slijedećih osam bita datoteke pokazivat će o kojemu je znaku riječ prema standardnoj ASCII tablici. Tako će 4% znaka u ulaznoj datoteci zahtijevati 5+8=13 bita.

Temeljna ideja je pridijeliti često korištenim znakovima manje bitova, a rijetko korištenim znakovima više bitova. U opisanom primjeru, srednji broj bita je $0.96 \times 5 + 0.04 \times 13 = 5.32$. Drugim riječima, ukupni odnos sažimanja: **8 bita/5.32 bita** ili oko **1.5:1**.

Huffmanovo kodiranje odvodi ova ideju do krajnosti. Znakovima koji se pojavljuju najčešće (razmak, točka, zarez...) pridjeljuje se jedan ili dva bita. Vrlo rijetki znakovi (I, @, #, \$, %) zahtijevaju više bitova. Matematički situacija je idealna ako je broj bita za svaki znak **proporcionalan logaritmu vjerojatnosti pojavljivanja znaka**.

Inteligentna osobina *Huffmanovog* kodiranja je način kojim se zajedno pakiraju kodovi promjenjive dužine. Zamislimo primanje serijskog niza podataka sastavljenog od jedinica i nula. Ako je svaki znak sastavljen od osam bita možemo odijeliti jedan znak od drugog tako da ih n rastavljamo na dijelove od 8 bita. Zamislimo sada niz podataka kodiran *Huffmanovim* postupkom u kojem svaki znak ima promjenjivi broj bita.

Pitanje je kako ćemo ih odijeliti?

Odgovor je u načinu odabira *Huffmanovih* kodova koji omogućavaju točno odjeljivanje. Primjer će objasniti princip rada.

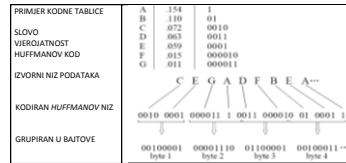
Ozren Bilan

7

Slika pokazuje pojednostavljenu *Huffmanovu* shemu kodiranja koja pokazuje slova od A do G u izvornom nizu podataka s vjerojatnošću pojavljivanja. Budući da je slovo A najčešće predstavljeno je kodom sa samo jednim bitom, kod: 1. Drugi najčešći znak je B, predstavljen je sa dva bita, kod: 01. To se nastavlja do slova koje se najmanje pojavljuje G, kojemu je pridijeljeno šest bita 000011. Na slici je prikazano kako su kodovi promjenjive dužine sortirani u grupama od osam bitova što je standard za računalni primjeru.

Pri dekompresiji svih osam grupa bitova redaju se jedna za drugom tvoreći serijski niz jedinica i nula. Pogledamo ih pažljivo kroz tablicu na slici, primijetiti čemo kako se svaki kod sastoji od dva dijela: *broj nula prije i jedinice i opcijni binarni kod nakon jedinice*. To omogućava binarnim nizovima podataka dijeljenje u kodove bez potrebe za granicama ili markerima između kodova. Program za dekompresiju promatra niz jedinica i nula dok se ne formira ispravan kod, pa onda zapisuje s traženjem novog znaka. Način formiranja kodova jamči da ne postoji dvosmislenost pri odjeljivanju.

Složniji postupak *Huffmanovog* pristupa naziva se **aritmetičko kodiranje** i koristi individualni kod usklađen s vjerojatnošću pojavljivanja znaka. Sažimanje je tada do 10%. *Run-length* kodiranje primjenjuje se nakon *Huffmanovog* postupka, a takvi algoritmi su vrlo složeni.



Huffmanovo kodiranje. Kodna tablica pridjeljuje svakom od sedam slova iz primjera binarni kod promjenjive dužine temeljem vjerojatnosti pojavljivanja. Izvorni niz podataka sastavljen od 7 znakova pretvara se iz tablice u *Huffmanov* kod. Budući da je svaki *Huffmanov* kod različite dužine, binarni podatke treba preprižuriti u standardnih 8 bita pri pohrani i prijenosu.

Ozren Bilan

8

Delta kodiranje

U znanosti i matematici, Grčko slovo *delta* (Δ , δ) koristi se za oznaku **promjene varijable**. Izraz *delta* kodiranje, označava nekoliko tehnika koje **pohranjuju podatke kao razliku sukcesivnih uzoraka** umjesto neposredne pohrane samih uzoraka. Slika pokazuje primjer kako se to postiče. Prva vrijednost *delta* kodirane datoteke identična je prvaj vrijednosti izvornih podataka, a sve iduće vrijednosti kodirane datoteke jednake su razlici (delta) između pripadajuće vrijednosti ulazne datoteke i prethodne vrijednosti.

IZVORNI NIZ PODATAKA	17	19	24	24	21	15	10	89	95	96	96	95	94	94	95	93	90	87	86	86	...	
KODIRAN DELTA NIZ	17	2	5	0	-3	-6	-5	79	6	1	0	0	-1	-1	0	1	-2	-3	-3	-1	0	...

Primjer *delta* kodiranja. Prva vrijednost kodirane datoteke je ista kao i prva vrijednost izvorne datoteke. Nakon toga, svaki uzorak kodirane datoteke je razlika tekućeg i prethodnog uzorka izvorne datoteke.

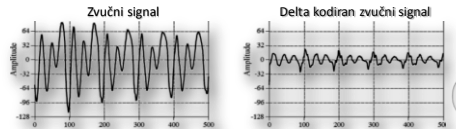
Ozren Bilan

9

Delta kodiranje se koristi ako postoje vrlo mala razlika susjednih vrijednosti izvornih podataka. To nije slučaj za ASCII tekst i tablice izvrsnih kode; međutim, to je vrlo uobičajeno ako datoteke predstavljaju zvučni signal. Slika a pokazuje odsječak zvučnog signala, digitiziran 8 bitno, gdje svaki uzorak ima vrijednost između -127 i 127. Slika b pokazuje *delta* kodirani inačicu istog signala. *Delta* kodirani signal ima nisku amplitudu od izvornog signala. Drugim riječima, *delta* kodiranje povećava vjerojatnost da će vrijednost svakog uzorka biti blizu nule, a smanjuje vjerojatnost da će biti jednako od nule. Takva vjerojatnost je ključna za funkcioniranje *Huffmanovog* kodiranja. Ako se izvorni signal ne mijenja ili se mijenja pravilnije, *delta* kodiranje imać će za posljedicu niz uzoraka iste vrijednosti.

Delta kodiranje praćeno *Huffmanovim* postupkom i/ili *run-length* kodiranjem je opća strategija sažimanja signala. Ideja korištena za *delta* kodiranje može se razviti u složeniju tehniku **Linearno Prediktivno Kodiranje** ili LPC.

Kako bi shvatili LPC, zamislimo da je prvih 99 uzoraka ulaznog signala kodirano i da obrađujemo uzorak broj 100. Zapitajmo se: temeljem poznavanja prvih 99 uzoraka, koja je najvjerojatnija vrijednost stotog uzorka? Primjenom *delta* kodiranja, najvjerojatnija vrijednost stotog uzorka bit će ista kao prethodna vrijednost, uzorak 99. Ta očekivana vrijednost koristi se kao referencijska vrijednost za kodiranje uzorka 100. Dakle, razlika između uzorka i očekivane vrijednosti postavlja se u kodiranu datoteku. LPC dalje poboljšava ovaj postupak boljom estimacijom najvjerojatnije vrijednosti. To se izvodi analiziranjem nekoliko prethodnih uzoraka, umjesto samo pretposljednog uzorka. Algoritmi LPC slični su rekurzivnim filtrima, dobivenih korištenjem *z*-transformacije i drugih viših matematičkih postupaka. LPC kodiranje se najviše koristi pri obradi govornih signala.



Ozren Bilan

10

LZW sažimanje

LZW postupak sažimanja nazvan je po autorima *Lempel, Ziv i Welchu*. To je najraširenija tehnika za opći postupak sažimanja podataka zbog jednostavnosti i svestranosti. LZW može sažeti tekst, izvršne kodne tablice, signale na polovinu izvorne veličine. LZW je primjenjiv na redundantne datoteke kao što su tablice, izvorni kodovi i pribavljeni signali. Tipični odnos sažimanja je 5:1.

Postupak LZW sažimanja koristi **kodnu tablicu**, prikazanu na slici. Uobičajeno je sastavljena od 4096 članova. Prikazani primjer LZW kodiranih podataka sastoji se od 12 bitnog koda koji se odnose na jedan element kodne tablice. Dekompresija se postiže tako što se svaki kod sažete datoteke uspoređuje s kodnom tablicom kako bi se odredilo koji znak kod predstavlja.

U kodnoj tablici kodovi od 0-255 uvijek predstavljaju jedinstveni bajt ulazne datoteke. Tako ako koristimo samo prvih 256 kodova, svaki bajt izvorne datoteke pretvorit će se u 12 bita u LZW kodiranoj datoteci, što ima za posljedicu 50% veću datoteku. Za vrijeme dekompresije, svaki 12 bitni kod pretvara se preko kodne tablice u jedinstveni bajt informacije.

Ozren Bilan

11

PRIMER KODNE TABLICE	code number	translation
KODNI BROJ PRJEVOD	0000	0
	0001	1
	:	:
	0254	254
	0255	255
	0256	145 201 4
	0257	243 245
	:	:
	:	:
	4095	xxx xxxxxx
IZVORNI NIZ PODATAKA	123 145 201 4 119 89 243 245 59 11 206 145 201 4 243 245	
KODIRAN PREKO KODNE TABLICE	123 256 119 89 257 59 11 206 256 257	

Primjer tablice kodiranja pri sažimanju koja je temelj LZW postupka. Kodiranje nastaje identificiranjem niza bajtova u izvornoj datoteci koja postoji u kodnoj tablici. 12 bitni kod koji predstavlja niz pohranjuje se u sažetoj datoteci umjesto tog niza. Prvih 256 zapisa tablice odgovaraju jednoj vrijednosti bajta od 0 to 255, dok preostali zapisi odgovaraju nizovima bajtova. LZW algoritam je učinkovit način generiranja kodne tablice temeljem podataka koji se sažimaju. (kodna tablica na slici je pojednostavljeni primjer, a ne generiran LZW algoritmom)

Ozren Bilan

12

LZW postupak postiže sažimanje uporabom kodova od 256 do 4095 kako bi predstavio nizove bajtova. Tako, npr. kod 523 može biti predstavljen nizom od tri bajta: 231 124 234. Svaki put kad algoritam sažimanja naiđe na prvi niz u ulaznoj datoteci, u kodiranu datoteku upisuje se kod 523. Za vrijeme dekompresije, kod 523 prevodi se preko kodne tablice u izvorni niz od tri bajta. Učinkovitost sažimanja je viša što su duži nizovi predstavljeni jedinstvenim kodom i što se nizovi češće ponavljaju.

Pri ovom jednostavnom pristupu potrebno je savladati dvije poteškoće:

- (1) kako odrediti koji niz treba biti u kodnoj tablici i
- (2) kako omogućiti programu za dekompresiju rad s istom kodnom tablicom koju je koristio program za sažimanje.

LZW algoritam rješava ova problema.

Pri početku sažimanja LZW programom, kodna tablica sadržava samo prvih 256 zapisa, a ostatak tablice je prazan. To znači da su prvi kodovi, koji se upisuju u sažetu datoteku, jedinstveni bajtovi ulazne datoteke pretvoreni u 12 bitova. Kako kodiranje napreduje, LZW algoritam identificira nizove podataka koji se ponavljaju i dodaje ih u kodnu tablicu. Sažimanje započinje tek kada se niz susretne po drugi put. Pri tome je ključno što se nizovi ulazne datoteke ne upisuju u kodnu tablicu sve dok se ne upišu u sažetu datoteku kao individualni znakovi (kodovi od 0 do 255). To je vrlo važno jer nam omogućava da program za dekompresiju rekonstruira kodnu tablicu neposredno iz sažete datoteke, pa nije potrebno posebno prenositi kodnu tablicu.

U sljedećim vježbama objasnit ćemo razlike postupka sažimanja zvučnih datoteka.

Ozren Bilan

13

Objašnjenje zvučnog sažimanja postupkom najmanjeg kvadrata

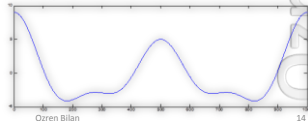
Prikazat ćemo temeljnu ideju sažimanja zvučnih podataka postupkom najmanjih kvadrata tako što ćemo zamijeniti izvornu zvučnu datoteku aproksimacijom sastavljenom od linearne kombinacije funkcija kosinusa. Metoda najmanjih kvadrata može se shvatiti kao postupak zamjene velikog skupa podataka s koeficijentima modela koji aproksimira podatke minimirajući razliku između početnih podataka i modela.

PRIMJER

Neka je zadana funkcija $f(t) = \cos(t) + 5\cos(2t) + \cos(3t) + 2\cos(4t)$.

Dijagram prikazuje funkcije $f(t)$ za $0 \leq t \leq 2\pi$ prikazan je plavom krivuljom. Podrazumijevamo da je zadan skup podataka od 1000 diskretnih vrijednosti funkcije $f(t)$ pravilno raspodijene unutar intervala $0 \leq t \leq 2\pi$. Podatke možemo u potpunosti interpolirati modelom **matrice A**:

```
t = linspace(0,2*pi,1000)';
b = cos(t) + 5*cos(2*t) + cos(3*t) + 2*cos(4*t);
A = [ones(size(t)),cos(t),cos(2*t),cos(3*t),cos(4*t)];
>> x=A\b
x =
    0.0000
    1.0000
    5.0000
    1.0000
    2.0000
>> plot(b)
```



>> plot(A)

Tada rješavamo linearni sustav Ax = b pomoću naredbe x=A\b.

Riješimo sustav opisanim postupkom

Komponente vektora koji predstavlja rješenje opisuju koeficijente početne funkcije. Neki koeficijenti su mnogo veći od ostalih pa aproksimiramo funkciju f primjenom aproksimacije najmanjih kvadrata koja izostavlja dijelove modela s koeficijentima koji imaju vrlo male vrijednosti. Tako npr. možemo postaviti izraz za model najmanjih kvadrata

$$A = [\cos(2^*t), \cos(4^*t)];$$

i riješiti pripadajući sustav najmanjih kvadrata x=A\b.

Ovaj model koristi samo dva koeficijenta za opis skupa od 1000 podataka.

Podudarnost je zadovoljavajuća, a prikazana je crvenom krivuljom. Dijagram dobijemo naredbom:

plot(t,b,'b',t,A*x,'r');

funkcija kosinus oscilira pravilnom frekvencijom. Višekratnici t u primjeru odgovaraju različitim frekvencijama jer što je veći višekratnik t, viša je frekvencija osciliranja. Metodom najmanjeg kvadrata odredili smo najbolju aproksimaciju podataka primjenom samo dvije frekvencije.

>> plot(t,b,'b',t,A*x,'r')

Ozren Bilan

Usporedba digitalnih formata WAV, WMA, AAC, MP3, OGG sigma delta mjernim sustavom

Postoji više digitalnih formata digitalnih audio signala, ali ako se radi o obradi primjenom računala trenutno su najzastupljeniji oblici:

- WAV vawe format – bez sažimanja
- AAC MPEG-2 (Advanced Audio Coding)
- WMA Windows Media Audio
- MP3 MPEG Audio Layer-3
- OGG format Ogg Vorbis

Da bi se, ETF moduliran, zapis na CD-u pripremio za obradu računalom, potrebno je prvo napraviti snimak u formatu koje računalo razumije. To je WAV format. Da bi se CD snimak pretvorio u WAV, potrebno ga je provući kroz *ripper*. To su programi koji pretvaraju, ETF modulirani, CD format u *.wav oblik. WAV datoteka se zatim može lako transformirati, dostupnim koderima, u MP3, WMA, AAC ili OGG oblik.

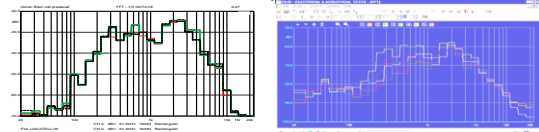
Interesira nas koji je od ovih oblika najbolji slušateljima? Analize subjektivnim ispitivanjem na *prosječnom audiolofskoj opremi* i analize mjerenjem sigma delta sustavom, pokazale su da pri protoku 96kb/s, što je približno odgovara CD kakvoći, nema razlike između navedena 4 formata, a sve objektivne razlike ako postoje, potpuno su subjektivno zanemarive.

Ozren Bilan

16

Za subjektivno ispitivanje isabrane je snimka u živo *Komorna orkestra Međunarodnog festivala u Bressi i Bergamu*. Odabrana je skladba *Introduzione V (Andante) Pietra Antonio Locatella* s CD Fone *glavni* CDC 085, pod nazivom *Sei Introduzioni Teatrali*. Snimka je odabrana zbog toga jer je snimljena i izvedena u Teatru *Grande di Brescia*, koja je sadržavala originalnu akustiku, kao u vrijeme nastanka skladbe. Pri snimanju je korištena posebna mikrofonska tehnika uz tri ambijentalna mikrofona. Tim pristupom zahvaćeni su detalji akustike prostora i zračenosti orkestra. Korišteni su mikrofoni Neumann U47 u OMNI položaju, smješten u orkestru za direktni glavni zvuk. Drugi par mikrofona Neumann U67 u KARDIOIDOM položaju smješten su ispred orkestra, a jedan mikrofoni Neumann U47 prima je ambijentalne informacije. Svi mikrofoni su korišteni u cijevna pretpočalica, pojedinačni mikrofoni pretpočalica NAGRA. Signali su proslijeđeni na stol za miješanje AMEKs u AD pretvorenič LHM NUMBUD. Signal je sniman na modulariziranom SONY DAC 1000 uređaju s uzorkovanjem od 44.1 kHz. Signal izabranog uzorka s fone *glavni* CDC 085 kompaktni diska, konvertiran je u WAV vawe format, MP3 MPEG Audio Layer-3, AAC, WMA Windows Media Audio i OGG format Ogg Vorbis. Za konverziju su korišteni izvorni kodovi. Potom su svi formati prebačeni na CD medij CDR uređajima programom NERO.

Korištenjem CD i CD-ROM reproduktora, izvršena su subjektivna testiranja zvuka profesionalnim slušalicama, kao i sustavom za reprodukciju zvučnicima, u posebnoj akustički obradenoj prostoriji, u skladu s preporukama IEC-a. U testovima je učestvovalo 20 slušatelja i statistički se nije mogla ustanoviti razlika između četiri formata; odnosno svi formati imali su podjednaku preferenciju slušatelja sa statističkom razlikom +F/2%, što je zanemarivo. Primjenom mjernog, delta-sigma sustava, izmjerene su vršne maksimalne i minimalne oktavne vrijednosti signala kodiranih arhiva što je prikazano na slikama. Svi protoci pri kodiranju su 96 kb/s. Analizom se može uočiti, kako su razlike veće pri minimalnim vrijednostima signala, nego pri maksimalnim i kako većina sustava ima vršne vrijednosti znatno veće od izvornih, kao i minimalne vrijednosti signala više od izvornih



Maksimalne i minimalne vrijednosti signala, kodiranih arhiva, u posebnoj akustički obradenoj prostoriji, u skladu s preporukama IEC-a. U testovima je učestvovalo 20 slušatelja i statistički se nije mogla ustanoviti razlika između četiri formata; odnosno svi formati imali su podjednaku preferenciju slušatelja sa statističkom razlikom +F/2%, što je zanemarivo. Primjenom mjernog, delta-sigma sustava, izmjerene su vršne maksimalne i minimalne oktavne vrijednosti signala kodiranih arhiva što je prikazano na slikama. Svi protoci pri kodiranju su 96 kb/s. Analizom se može uočiti, kako su razlike veće pri minimalnim vrijednostima signala, nego pri maksimalnim i kako većina sustava ima vršne vrijednosti znatno veće od izvornih, kao i minimalne vrijednosti signala više od izvornih

Ozren Bilan

MPEG Layer-3 je shema perceptualnog audio kodiranja koja analizira audio signal i primjenjuje psihosustički model, upotrebljavajući karakteristike ljudskog uha, pokušavajući zadržati izvornu kvalitetu signala. U glazbi, uslijed postupka maskiranja, ljudski organ sluha ne može istovremeno čuti glazne i tihe zvukove. One zvukove koje ljudski organ sluha ne može čuti, algoritam za sažimanje podataka odbacuje.

Naglasili smo da su podaci na CD disku snimljeni su s 1411 kb/s, a vrlo kvalitetan MP3 zapis je 192 kb/s ili čak 320 kb/s. 96kb/s odgovara CD kakvoći. Najveći broj MP3 snimljen je s 128 kb/s, što je visoka audio kvaliteta pri kojoj većina ljudi neće primijetiti razliku u izvornim CD diskom u reprodukciji preko računala.

WMA format bolji je pri nižim digitalnim protocima, dok je MP3 bolji za visoke protoke. Na vrlo niskim brzinama kao 16 Kb/s, WMA radi bolje od MP3, ali ni s jednim formatom ispod 64 kb/s nije dobra bas reprodukcija.

Dakle, optimalno je korištenje 96 Kb/s. Korištenjem AAC, MP3, OGG i WMA koda pri protoku od 96 kb, subjektivni testovi pokazuju da nema razlike između zvuka navedenih koda i WAV formata s *prosječnom računalom audiolofskom opremom*.



Ozren Bilan

18

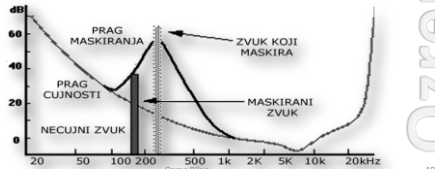
Verifikacija koncepta maskiranja i prikaz kodiranja MPEG-1 Layer-3 (MP3)

Sažimanje bez gubitaka

Zvučne datoteke se sažimaju bez gubitaka tako da pri dekompresiji dobivamo signal koji je potpuno identičan početnom signalu. Iako ima prednosti, ova vrsta sažimanje nije postigla veći popularnost kod korisnika jer za zvuk CD kvalitete (16 bita, 44.1 kHz) postit će mali stupanj sažimanja 30% do 50%. Tehnike sažimanje bez gubitaka se uglavnom razlikuju po brzini sažimanja i dekompresije dok nemaju uticaj na kvalitet sadržaja.

Sažimanje s gubicima

Zvuk koji je manje važan kodira se smanjenom preciznošću ili odbacuje. Kako bi se odredilo koje su informacije u zvučnom signalu manje važne, većina algoritama koristi *modificiranu diskretnu kosinusnu transformaciju* (MDCT). Frekvencijskim komponentama alokiraju se bitovi temeljem čujnosti. Čujnost frekvencijske komponente definira se tako da se prvo izračuna prag maskiranja ispod koje se pretpostavlja da je prema psihoakustičnom modelu zvuk izvan granice čujnosti ljudske percepcije.



Ozren Bilan

19

Neki algoritmi sažimanja se sa gubicima koriste LPC (Linear Predictive Coding). Pošto kod sažimanja se sa gubicima dolazi do slabljenja kvaliteta signala, ovakav tip sažimanja drži se neodgovarajućim za profesionalne aplikacije editiranja zvuka i visekanalnog snimanja. Međutim, u sažimanju se veoma pogodno za prijenos i pohranu zvučnih podataka.

Postupci zvučnih sažimanja

Postoji nekoliko algoritama sažimanja i postupaka pohrane zvučnog sadržaja:

MP3 (MPEG-1 Layer-3)

Sastoji se od psihoakustičnog modela, FFT analiza, hibridnih filtera, nelinearne kvantizacije, Huffman-ovog kodiranja, 2 kanala sa konstantnim ili promjenjivim brzinama bita od 32 do 256 kbps i koji je uspješno implementiran u moderne CD/DVD reproduktore, mobilne telefone,...

MP3 Pro

Razvili su ga Coding Technologies Laboratory, temelji se na principima formiranja MP3 formata, sastoji se od SBR (Spectral Band Replicator) tehnologije koja kodira visoka frekvencijska pojas od 10 do 15kHz sa veoma malom brzinom bita.

AAC (MPEG-2 Advanced Audio Coding)

Razvili su ga AT&T, Dolby, Fraunhofer IIS i Sony. ISO standard MPEG-2 koji se sastoji od osnovnih principa MP3 formata. uključujući psihoakustički model, hibridne filtere, skalabilne brzine sampliranja, 2 kanala sa brzinama od 48 do 576 kbps.

AAC Plus

Postoje od AAC formata, sastoji se od SBR-a sa brzinom bita do 100 kbps.

WMA (Windows Media Audio)

Razvio ga je Microsoft, sastoji se od DRM (Digital Rights Management), CBR i VBR. WMA kodiranjem digitalnog zvučnog signala bez gubitaka (brzine bita 2:1 do 3:1), WMA profesionalno kodiranje visekanalnog zvučnog signala (128 do 768 kbps) i WMA za kodiranje govora (od 4 do 20 kbps).

VOX (Voice Quantization File)

nastao je temeljem MP3 od strane NINT Human Interface Laboratories i Yamaha, kompleksnije kodiranje, 25-35% bolja sažimanja u odnosu na MP3 sažimanje.

Ozren Bilan

20

Verifikacija koncepta maskiranja

MATLAB nam omogućava specifikaciju velikog broja i generiranje pripadajućih zvučnih datoteka za ispitivanje. Opatim postupkom koristeći verifikaciju koncepta audio maskiranja pri postavljanju MP3.

PROGRAM

Sljedeći program generira datoteke koje prikazuju koncept audio maskiranja.

```
f=44100; % sampling frequency
nb=16; % 16-bit sample
sig1=0.5*sin(2*pi*1000*44100*(1:1+44100));
% 2000 Hz, 1 zvučno signalo
sig2=0.5*sin(2*pi*1250*44100*(1:1+44100));
% 2500 Hz, 1 zvučno signalo
fs=44100; % sampling frequency
nb=16; % 16 bit sample
sig1=0.5*sin(2*pi*1000*44100*(1:1+44100));
% 2000 Hz, 1 osvoj
sig2=0.5*sin(2*pi*1250*44100*(1:1+44100));
% 2500 Hz, 1 osvoj
sig3=0.5*sin(2*pi*1000*44100*(1:1+44100));
% 2000 Hz tonovi
jednakoj osvoj
sig4=0.5*sin(2*pi*1000*44100*(1:1+44100));
% 2000 Hz je 2008 jodi
2500 Hz
sig5=0.5*sin(2*pi*1000*44100*(1:1+44100));
% 2000 Hz je 4008 jodi
40250 Hz
```

```
wavwrite(sig1,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1.wav');
wavwrite(sig2,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig2.wav');
wavwrite(sig3,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig3.wav');
wavwrite(sig4,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig4.wav');
wavwrite(sig5,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig5.wav');
wavwrite(sig1+sig2,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2.wav');
wavwrite(sig1+sig3,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig3.wav');
wavwrite(sig1+sig4,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig4.wav');
wavwrite(sig1+sig5,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig5.wav');
wavwrite(sig1+sig2+sig3,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig3.wav');
wavwrite(sig1+sig2+sig4,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig4.wav');
wavwrite(sig1+sig2+sig5,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig5.wav');
wavwrite(sig1+sig2+sig3+sig4,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig3+sig4.wav');
wavwrite(sig1+sig2+sig3+sig5,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig3+sig5.wav');
wavwrite(sig1+sig2+sig3+sig4+sig5,fs,nb,
'\\Users\Ozren\Documents\MATLAB\sig1+sig2+sig3+sig4+sig5.wav');
```

Ozren Bilan

21

Reproduciraj svi tri valna oblika.

Potvrdi slušanjem da se pri izvođenju test2.wav jedva detektira signal 2500 Hz koji je 40dB ili 0.1 puta slabiji i da u test3.wav, nije moguće detektirati signal 2500 Hz koji je 40dB ili 0.01 puta slabiji.

Prag maskiranja je subjektivan i mijenja se od osobe do osobe. Na zadatku u vježbi eksperimentiraj s odnosom signala pa zaključi koliko ti je uho osjetljivo

Nadopuni program tako da grafički prikaže sve valne oblike signala i do 5 i u testovima od 1 do 3. Poslušaj zvučnicima ili slušalicama signale.

Komentiraj temeljem nacrtanih valnih oblika i onoga što se čuje. Promjeni amplitudu signala od 2000 Hz na više i na niže; što primjećuješ? Na koji frekvenciji i koji razini čuješ razliku dva signala?

MPEG audio sažimanja / MPEG-1 Layer-3 (MP3)

The Motion Picture Experts Group (MPEG) radna grupa osnovana 1988. godine koja definira standarde za video i audio sažimanja. Objavljen 1993. godine od strane International Standards Organization / International Electrotechnical Commission (ISO/IEC). MPEG-1, ISO/IEC 11172 standard uključuje specifikacije za 1-2 Mbps video sažimanja i tri sloja za audio sažimanja. Termin MP3 je obično korišten kao referenca za MPEG-1 Layer-3 specifikaciju za kodiranje zvučnog signala. MP3 standard određuje procese dekodiranja, formata i strategiju kodiranja zvučnog signala. Algoritam je razvio *Fraunhofer Institut*. Test je pokazao vrlo visoku kvalitetu i pri sažimanju 6:1.

Ljudsko uho ima ograničenu frekvencijsku selektivnost pa se čini spektar dijeli na kritične pojaseve. Predstavljaju moć razlučivanja ljudskog uha u funkciji frekvencije. Zbog toga kritična točka maskiranja šuma bilo koje frekvencije zavisi o zvučnom signalu u kritičnom pojasu te frekvencije.

Pri sažimanju transformiramo zvučni signal u frekvencijsko područje pa podijelimo spektar u podpojase koji odgovaraju kritičnim pojasi ljudskog sluha. Potom kvantificiramo svaki podjase temeljem čujnosti šuma koji se nalazi unutar pojasa. Za optimalno sažimanje, svaki pojas bi se mora kvantificirati minimalnim brojem razina kako bi šum bilo nečujnan.

MPEG audio kodiranje i dekodiranje

Ulazni audio signal prolazi kroz filteri slog koji dijeli ulazni signal na više različitih podpojaseva. Ulazni zvučni signal prolazi kroz psihoakustični model koji određuje odnos za svaki podjase. Blok za dodjeljivanje, koristi odnos signal/masker kako bi pravilno samplirao i dodijelio broj kodiranih bitova potrebnih za određivanje svih podjasnih signala sa ciljem minimizacije šuma. Na kraju, posljednji blok uzima kvantizirani audio uzorak i formira bitove koje je moguće dekodirati. Dekoder jednostavno vrši oburno formiranje, zatim rekonstruira kvantizirane vrijednosti svih podpojasa u vremensko područje audio signala.

MPEG audio standard ima tri sloja za sažimanje. Layer I formira najjednostavniji algoritam, a Layer II i Layer III su nastavci koji koriste neke elemente iz Layer I. Svaki od ovoj slojeva poboljšava karakteristike sažimanje uz veću složenost kodera i dekodera.

Ozren Bilan

23

LAYER I

Layer I algoritam koristi osnovni filteri niz koji se može naći u svim ostalim slojevima. Ovaj niz dijeli signal na 32 jednaka frekvencijska pojasa. Prvo, 32 frekvencijska pojasa ne odgovaraju kritičnim pojasi ljudskog uha. Filteri niz na izlazu ima 32 izlazna signala. Layer I algoritam zajedno grupira 12 uzoraka za svaki od 32 pojasa.

Svakoj grupi od 12 uzoraka se dodjeljuje jedan bit i (ako bit nije nula) faktor skaliranja. Dodjeljivanje bita određuje broj bitova koji predstavljaju svaki uzorak. Layer I koder formira 32 grupe po 12 uzoraka (ukupno 384 uzorka) u jedan okvir. Dored zvučnih podataka, svaki okvir sadrži zaglavje i CRC cyclic redundancy code provjeru i dodatne podatke.

LAYER II

Layer II algoritam je jednostavno proširenje Layer I. Koder dijeli vrijednost faktora skaliranja između dve ili tri grupe samo u dva slučaja: (1) kada su vrijednosti faktora skaliranja izdvojeno blizu (2) kada koder očekuje da će privremeno maskiranje šuma sakriti izobličena. Layer II algoritam također poboljšava karakteristike Layer I reprezentacijom akcija bite, vrijednostima faktora skaliranja i korištenjem učinkovitijeg koda.

LAYER III

Layer III algoritam je mnogo finiji pristup. Iako je temeljen na istom filteri skom nizu, Layer III kompenzira nedostatak filteri skog niza tako što obrađuju izlaz *modificiranom diskretnom kosinusnom transformacijom* (MDCT).

Ozren Bilan

24

