

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

Teorija Informacij in sistemov, predavanje 4

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

Uroš Lotrič

Univerza v Ljubljani,
Fakulteta za računalništvo in informatiko

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

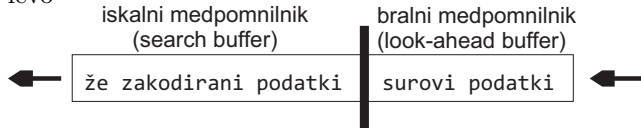
3.7.3
MPEG

- ▶ Do zdaj smo obravnavali stiskanje na osnovi verjetnosti
- ▶ Druga možnost je stiskanje na osnovi slovarja
- ▶ v besedah se pojavljajo vzorci, konstrukcija slovarja na osnovi teh vzorcev
- ▶ ne uporablja v naprej znanih verjetnosti za znake
- ▶ **kodirnik** med branjem niza gradi slovar in zapisuje reference na nize v slovarju
- ▶ **dekodirnik** med branjem kodnih zamenjav rekonstruira slovar in znake
- ▶ kodirnik in dekodirnik sprotno gradita slovar

- ▶ enostavna ideja, 1977; izkaže se, da se v limiti, ko je besedilo dolgo, približujemo entropiji besedila
- ▶ gre torej za univerzalni algoritem za stiskanje
- ▶ kodirnik:
 - ▶ preiskuje že poslana besedila, da poišče najdaljši niz, ki se je ponovil
 - ▶ namesto niza pošlje referenco na niz
 - ▶ idealno: pregledovanje celotne zgodovine
 - ▶ praksa: pregledovanje nazaj in naprej je omejeno

► kodirnik

- uporablja drseča okna, znaki se premikajo iz desne na levo



- enkoder bere znake v bralnem medpomnilniku in išče podobne nize v iskalnem medpomnilniku
- referenca (kodiranje) je podano kot trojček
 - odmik - razdalja do začetka enakega podniza v medpomnilniku
 - dolžina enakega podniza
 - naslednji znak
- Primeri: (5, 3, F), (0, 0, A) - ni ujemanja, (5, 9, R) - ujemanje se nadaljuje še v pregledovalni medpomnilnik

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

► Kodiranje niza GORI-NA-GORI-GORI.

kodne zamenjave			
0	0	G	G
0	0	O	O
0	0	R	R
0	0	I	I
0	0	-	-
0	0	N	N
0	0	A	A
3	1	G	-G
8	4	G	ORI-
5	3	.	ORI.

► Dekodiranje: sledimo kodnim zamenjavam

- ▶ malo predelan Lempel-Ziv
- ▶ uporabljaja pare (odmik, dolžina)
- ▶ če ni ujemanja, zapiše kar znak
- ▶ dve kodni tabeli:
 - ▶ tabela za znake in dolžine ima 285 simbolov - $[0..255]$ za osnovne znake, 256 konec bloka, 257 - 285 kodira dolžine. Ista koda z 0-5 dodatnimi biti za več dolžin: 257+0: 3, 265+1: 11-12, ... 281+5: 131-162. Kodne zamenjave brez dodatnih bitov se zakodira s Huffmanom (trik, da je drevo manj razvejano)
 - ▶ tabela odmikov (končni medpomnilniki - 32k). Kodiranje: 5 bitni enakomerni kod + dodatni biti (0-13)

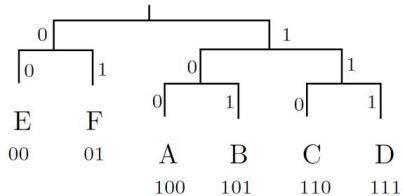
- ▶ niz znakov se razdeli na bloke, vsak blok se kodira na enega od treh načinov
 - ▶ brez stiskanja - osnovni znaki se prepisejo (blok omejen na 64k)
 - ▶ stiskanje s statičnim Huffmanom - verjetnosti podane v naprej, Huffmanovo drevo ni zakodirano v bloku
 - ▶ stiskanje s Huffmanom - verjetnosti so izračunane za blok - Huffmanovo drevo je zakodirano v bloku (počasnejše od prejšnjega, bolj stisnjeno)
- ▶ blok ima glavo: 1 bit - zadnji/ni zadnji blok + 2 bita - tip stiskanja + pri režimu 3 še Huffmanovo drevo

Huffmanovo drevo načeloma ni enolično definirano. Kako ga kodirati?

- ▶ Kanonični Huffmanov kod
 - ▶ znake razvrstimo najprej po dolžinah kodnih zamenjav in nato po abecedi
 - ▶ prvi simbol ima same ničle
 - ▶ vsakemu naslednjemu znak dodelimo naslednjo binarno kodo
 - ▶ če je kodna zamenjava daljša od binarne kode števila, na koncu pripnemo ničlo
- ▶ Kanonično drevo (izgled):
 - ▶ krajše kodne zamenjave na levi (0), daljše na desni (1)
 - ▶ če ima več znakov iste dolžine kodnih zamenjav, so na levi (0) tisti, ki so prej po abecedi

- ▶ Primer:
 $\{A, B, C, D, E, F\}, P = \{0.11, 0.14, 0.12, 0.13, 0.24, 0.26\}$

- ▶ dolžine kodnih zamenjav: $\{3, 3, 3, 3, 2, 2\}$
- ▶ dodelitev kodnih zamenjav glede na zgornja pravila



- ▶ na ta način dosežemo, da je treba kodirati samo dolžine kodnih zamenjav - lahko zelo učinkovito

- ▶ osnovni slovar je definiran
- ▶ algoritem (izpisuje v niz kodnih zamenjav)

$N = "$

ponavljaj:

preberi naslednji znak z

če je $[N, z]$ v slovarju:

$N = [N, z]$

drugače:

izpiši indeks k niza N

dodaj $[N, z]$ v slovar

$N = z$

izpiši indeks k niza N

3.5.3 Kod LZW: primer - kodiranje

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

- osnovni slovar (levo) in gradnja slovarja (desno)

indeks	niz	indeks	niz
1	-	9	GO
2	.	10	OR
3	A	11	RI
4	G	12	I-
5	I	13	-N
6	N	14	NA
7	O	15	A-
8	R	16	-G
		17	GOR
		18	RI-
		19	-GO
		20	ORI
		21	I.

- kodiranje niza

G	O	R	I	-	N	A	-	GO	RI	-G	OR	I	.
4	7	8	5	1	6	3	1	9	11	16	10	5	2

► med branjem rekonstruiramo slovar

► algoritem (izpisuje osnovni niz):

preberi znak k

v slovarju poišči niz N , ki ustreza indeksu k

izpiši N

$N_{star} = N$

ponavlaj:

preberi znak k

v slovarju za indeks k poišči niz N

izpiši N

v slovar daj $[N_{star}, N(1)]$, $N(1)$ je prvi znak

$N_{star} = N$



3.5.3 Kod LZW: dekodiranje 2

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

- ▶ pri rekonstrukciji slovarja vedno zaostajamo za eno kodno zamenjavo
- ▶ primer težave: kodiranje in dekodiranje niza 'ABABABA'

				indeks	niz	indeks	niz
A	B	AB	ABA	1	A	3	AB
1	2	3	5	2	B	4	BA
						5	ABA

- ▶ pri rekonstrukciji pridemo do niza, ki ga še ni v slovarju.
 - ▶ začeti se mora z zadnjim nizom, ki je še v pomnilniku, saj ta še ni šel v slovar (AB)
 - ▶ zadnji vpis v slovar je torej sprožil prvi znak tega niza (A)
 - ▶ skriti niz je torej ABA.

- ▶ popravljeni algoritem (izpisuje osnovni niz):
 - preberi indeks k
 - v slovarju poišči niz N , ki ustreza indeksu k
 - izpiši N
 - $N_{star} = N$
 - ponavljaj:
 - preberi indeks k
 - če je k v slovarju:
 - v slovarju za indeks k poišči niz N
 - drugače:
 - $N = [N_{star}, N_{star}(1)]$
 - izpiši N
 - v slovar dodaj $[N_{star}, N(1)]$
 - $N_{star} = N$

- ▶ najdemo ga v GIF
- ▶ slabost: velik slovar, vsega ne rabimo
- ▶ LZW doseže optimalno stiskanje - se približa entropiji
 - ▶ naj bo n dolžina iskalnega medpomnilnika
 - ▶ najdaljši podniz v medpomnilniku, ki je enak nizu v bralnem medpomnilniku, označimo z s_n^{\max}
 - ▶ Da se pokazati, da v limiti $n \rightarrow \infty$ velja

$$\frac{\log n}{s_n^{\max}} = \frac{1}{n} H(X_1, \dots, X_n)$$

- ▶ LZW doseže doseže optimalno stiskanje - se približa entropiji

- ▶ kodiramo niz sestavljen iz znakov A, B

ABAB	BABA
BABB	ABAB

$$s_4^{\max} = 3 \quad s_4^{\max} = 3$$

$$s_4^{\max} = 2$$

- ▶ vzemimo, da je $n = 1024 = 2^{10}$
- ▶ A in B prihajata naključno, $P = \{1/2, 1/2\} \rightarrow H = 1$
 $\rightarrow \log 1024/s_n^{\max} \approx 1$
 - ▶ Pričakujemo lahko, da je prihajajočih $s_n^{\max} = 10$ znakov že v slovarju, saj je verjetnost za niz 10 znakov 2^{-10} , različnih nizov pa je 2^{10} in velja $2^{10} \cdot 2^{-10} = 1$
 - ▶ za kodiranje niza 10 znakov potrebujemo 10 bitov \rightarrow vseeno če ne kodiramo
- ▶ $H = 1/2 \rightarrow s_d^{\max} = 20$: še vedno pošljemo 10 bitov za indeks, s tem opišemo 20 znakov \rightarrow idealno stiskanje glede na entropijo.

- ▶ Ena najstarejših tehnik stiskanja
- ▶ BMP, TIFF, PCX
- ▶ enostavna izvedba
- ▶ večinoma se uporablja, če so podatki samo dveh tipov č/b, 1/0
- ▶ izkorišča dejstvo, da se (na slikah) določeni podatki ponavljajo
- ▶ namesto originalnih podatkov se shranjuje dolžina verige: aaaabbc = 4a2b1c
- ▶ težava, če se podatki ne ponavljajo

3.6 Verižno kodiranje = run length encoding 2

Teorija
Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

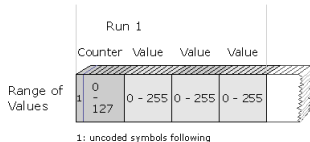
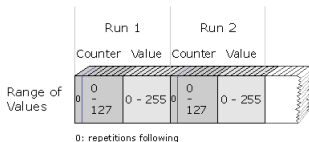
3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

- ▶ običajno kombinacija direktnega kodiranja in kodiranja RLE

- ▶ 8-bit RLE



- ▶ lahko tudi poseben znak za kodiranje RLE: kontrolni znak+število+vsebina
- ▶ ITU-T4: faksiranje sporočil: dolžine črnih in belih črt so ločeno zakodirane s Huffmanom. Vedno se izmenjujejo bela in črna
- ▶ Deflate: dolžine kodnih zamenjav, s katerimi se opisuje Huffmanovo drevo, so kodirane kot RLE

- ▶ faksiranje sporočil
 - ▶ bela in črna barva
 - ▶ vrstica se vedno začne z belo piko
 - ▶ ločljivost: H: 8/mm, V: 3,85/mm,
 - ▶ 1728 točk v vrstici
 - ▶ za list A4 210 x 297 = 1,92 Mbit, pri 9600 bit/s -> 200 s
- ▶ kodiranje po standardu ITU-T4
 - ▶ primer:

		x	x	x	x	x		x	x	x	x			
--	--	---	---	---	---	---	--	---	---	---	---	--	--	--

(2,b)(5,č)(1,b)(4,č)(3,b) → 2, 5, 1, 4,
 - ▶ ločena koda za dolžine belih in črnih pik
 - ▶ statični Huffman na podlagi kopice dokumentov
 - ▶ EOL v obeh kodih enak
 - ▶ 0 – 63: končne kodne zamenjave
 - ▶ 64+, 128+, 192+, ... 1728+ (korak 64)
 - ▶ primer: (65,b)(2,č)EOL → (64,b)(1,b)(2,č)EOL

Teorija

Informacij
in sistemov,
predavanje
4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

3.7.1 JPEG

3.7.2 MP3

3.7.3
MPEG

- ▶ Stiskanje brez izgub do sedaj
- ▶ Mnogo bolje se stiska, če dovolimo izgube
- ▶ Pishovizualni in psihoaktustični pristopi - pri stiskanju se upošteva kako dojemajo človekova čutila
- ▶ učinkovito pri slikah, zvoku in videu
- ▶ Kompresijsko razmerje: stisnjen dokument/osnovni dokument

$$R = C(M)/M$$

M - binarni zapis dokumenta, $C(M)$ stisnjeni binarni zapis.

- ▶ Tipična kompresijska razmerja:
 - ▶ Brez izgub (besedilo, koda, exe): 50 % – 75 %
 - ▶ Z izgubami (slike, zvok, video): 10 % in manj

Postopek kodiranja

- ▶ JPEG = Joint Photographic Experts Group
- ▶ priprava slike - shema $YC_R C_B$: svetlost + dve barvi, svetlost je bolj pomembna, barvna resolucija je zato običajno zmanjšana (4:2:2, 4:1:1)
- ▶ aproksimacija vsake od treh komponent z 2D diskretno cosinusno (ali valčno) transformacijo: slika se razdeli na bloke 8x8: enostavne strukture - majhne vrednosti, kompleksne - velike vrednosti: (primer: aproksimacija 1D stopnice s sinusi). Odrežemo majhne koeficiente.
- ▶ kvantizacija: oko je bolj občutljivo na majhne variacije barve na velikih površinah kot na velike variacije na majhnem prostoru - visoke frekvence (veliki koeficienti) so shranjeni manj natančno kot manjši
- ▶ kodiranje blokov s pomočjo entropije:

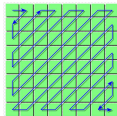
- ▶ predikcija vrednosti na podlagi sosdenjih točk (ideja - graf 1D)


Na sliki:

A	B
C	D

$$D = A + (B-A) + (C-A) = B+C-A$$

namesto absolutnih vrednosti dobimo majhne popravke



- ▶ RLE cik-cak po sliki 
- ▶ dolžine iz RLE kodirane: Huffman + aritmetični (odkar ni patenta, 5-7% bolj stisne)

Teorija
Informacij
in sistemov,
predavanje

4

ULotric

3.5
Stiskanje s
slovarjem

3.5.1
Lempel-Ziv

3.5.2
Deflate

3.5.3 LZW

3.6 Verižno
kodiranje

3.7
Stiskanje z
izgubami

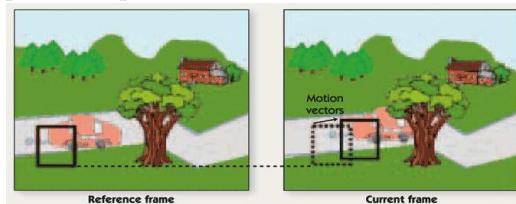
3.7.1 JPEG

3.7.2 MP3

3.7.3
MP3

- ▶ del MPEG standarda
- ▶ psihoakustični prestop
- ▶ faze:
 - ▶ Modified Discrete Cosinus Transform -> koeficienti
 - ▶ odstranitev za človeka neslišnih frekvenc
 - ▶ med močnimi zvoki ne slišimo šibkih zvokov in jih lahko odstranimo
 - ▶ da ne trpi kvaliteta zvoka pri hitrih prehodih se v naprej (ko je možno) pošiljajo potrebni podatki in se shranjujejo v predpomnilnik
 - ▶ stereo: če je L in R podobno, se pošilja vsota L+R in razlika L-R, za zelo nizke in zelo visoke frekvence težko ugotovimo od kje prihajajo -> pošilja se mono zvok z nekaj dodatnimi biti, da se da za silo rekonstruirati prostorskost
 - ▶ Huffman: na koncu, izveden na koeficientih MDCT

- ▶ MPEG = Motion Picture Experts Group
- ▶ več standardov (kodekov) MPEG1... MPEG4, MP3 je avdio za MPEG1
- ▶ osnova je JPEG
- ▶ štirje tipi okvirjev:
 - ▶ I: uvodno kodiranje: cela slika (JPEG)
 - ▶ P: prediktivno kodiranje: poslane so spremembe (v sliki (JPEG) + vektor premika (Huffman)). Če je razlik preveč se prenese I



- ▶ B: kodiranje iz predhodnega in naslednjega okvira (interpolacija)
- ▶ D: enako kot I, samo močno stisnjeni, zaradi previjanja filma