



Diplomski studij

Informacijska i komunikacijska
tehnologija:

Obradba informacija
Telekomunikacije i informatika

Višemedijske komunikacije

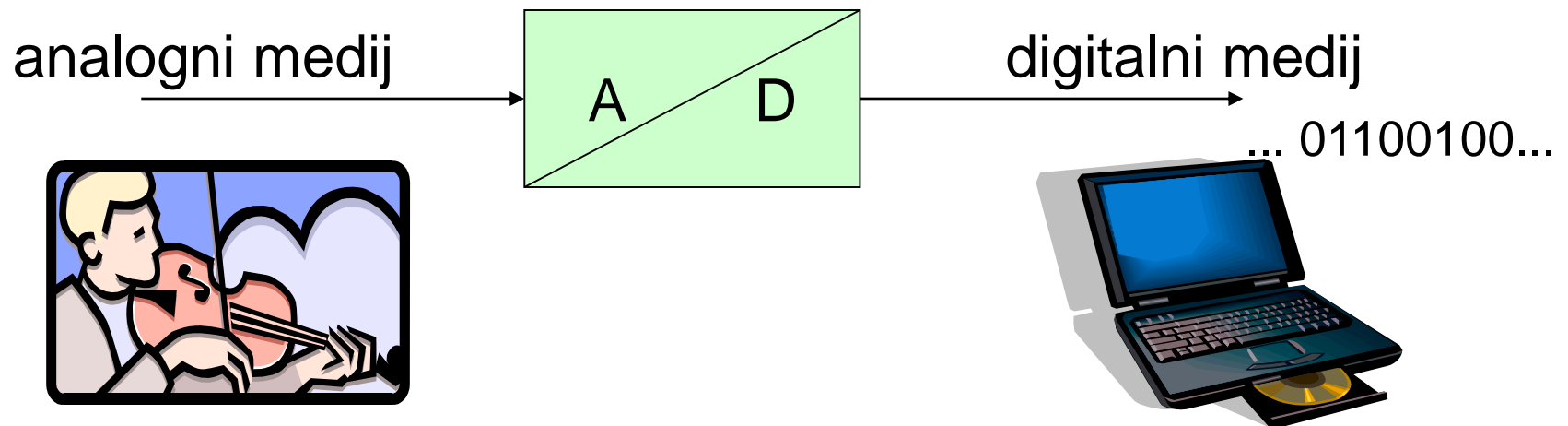
2.

Osnove kodiranja i kompresije

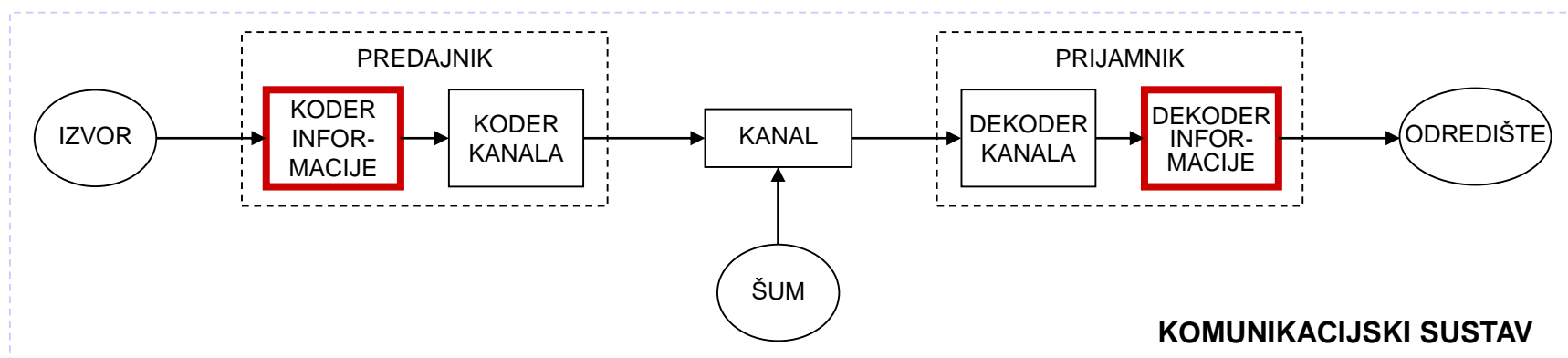
- ◆ Uvod, definicije
 - Što je višemedijski sadržaj?
 - Definicije, podjela metoda kompresije
 - Hibridno kodiranje
 - Kompresija u sklopu komunikacijskog sustava
- ◆ Entropijsko kodiranje
 - Karakteristike izvora informacije
 - Vrste kodova i njihova svojstva
 - Metode entropijskog kodiranja
 - ◆ Huffmanovo kodiranje
 - ◆ Aritmetičko kodiranje
 - ◆ Metode rječnika (LZ77, LZ78, LZW)
 - ◆ Metode skraćivanja niza (potiskivanje nula, slijedno kodiranje)
- ◆ Izvorno kodiranje
 - Osnovna svojstva i principi
 - Osnovne metode izvornog kodiranja
 - ◆ Kvantizacija
 - ◆ Poduzorkovanje
 - ◆ Transformacijsko kodiranje
 - ◆ Diferencijalno (predikcijsko) kodiranje
 - ◆ Potpojasno kodiranje
 - ◆ Kodiranje zasnovano na modelu

Što je (više)medijski sadržaj?

- ♦ analogni medij → uzorkovanje → kvantizacija → kodiranje/kompresija → digitalni medij
- ♦ medij se pretvara u digitalni oblik, pogodan za obradu, pohranu ili prijenos



- ◆ Kodiranje: dodjela kodnih riječi simbolima poruke
- ◆ Kompresija: kodiranje koje smanjuje broj bitova potreban za izražavanje poruke
- ◆ U jasnom kontekstu, koristimo ove pojmove kao sinonime
- ◆ Kompresija se vrši u koderu informacije

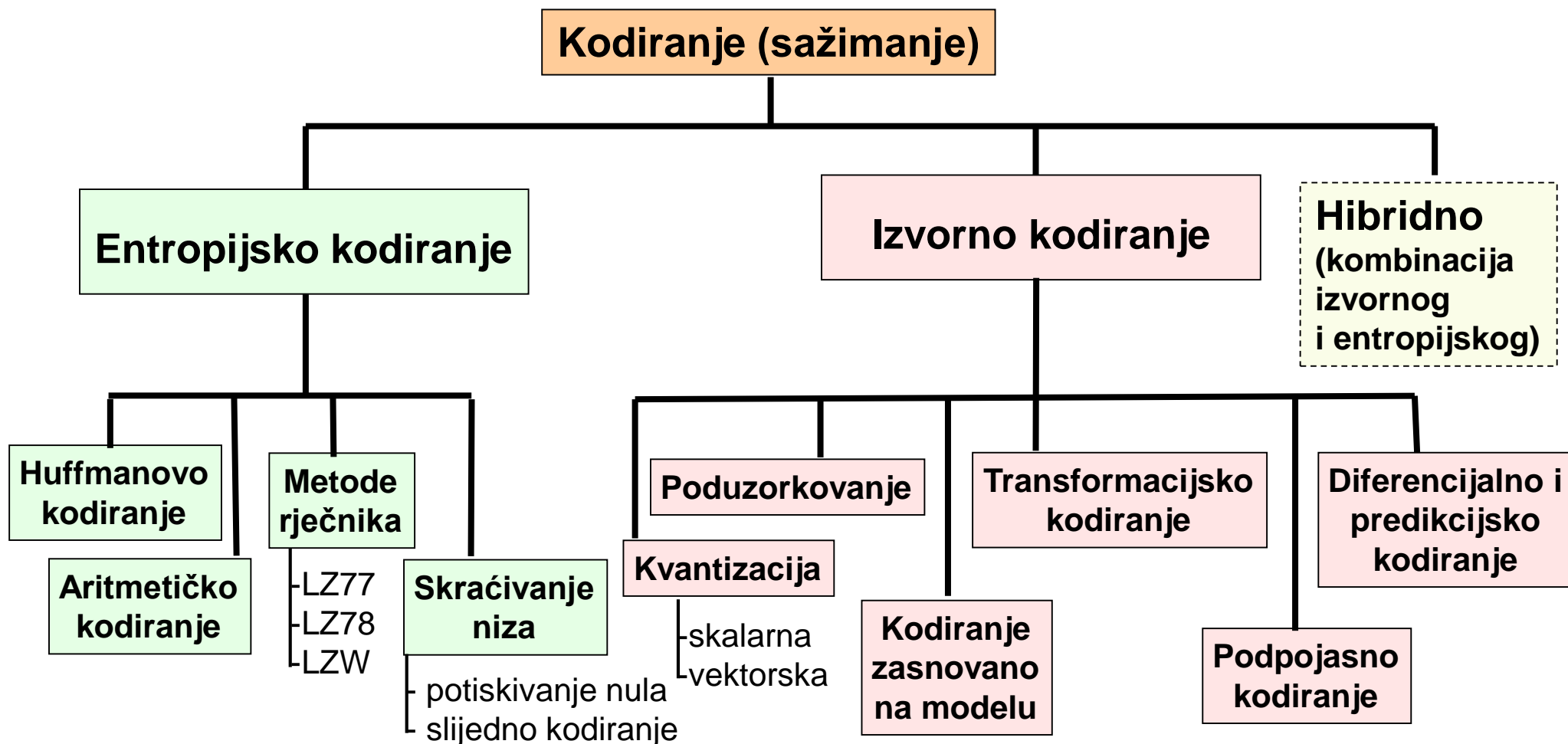


Zašto je kompresija moguća?

- ◆ Postupci kompresije se uglavnom temelje na redundanciji u podacima:
 - vremenska redundancija (npr. korelacija uzastopnih uzoraka audia)
 - prostorna redundancija (npr. korelacija susjednih elemenata slike - pixela)
 - spektralna (npr. korelacija između boja ili svjetline u slici)
 - percepcijska (ljudski percepcijski sustav; primjena: slika i zvuk)

- ◆ Kompresija **bez gubitaka**
 - Komprimirani podaci mogu se dekomprimiranjem rekonstruirati bez gubitka informacije (*reverzibilno*)
 - Primjene: npr. tekst, medicinske slike, satelitske snimke
- ◆ Kompresija **s gubicima**
 - Cilj je ili dobiti najbolju vjernost rekonstruiranih podataka za zadanu brzinu (bit/s) ili postići najmanju brzinu za zadanu granicu vjernosti
 - Primjene: npr. govor, slika, video
- ◆ Važan parametar je **omjer kompresije**
 - Omjer veličine komprimiranih i originalnih podataka, npr. 1:10

Klasifikacija postupaka kodiranja

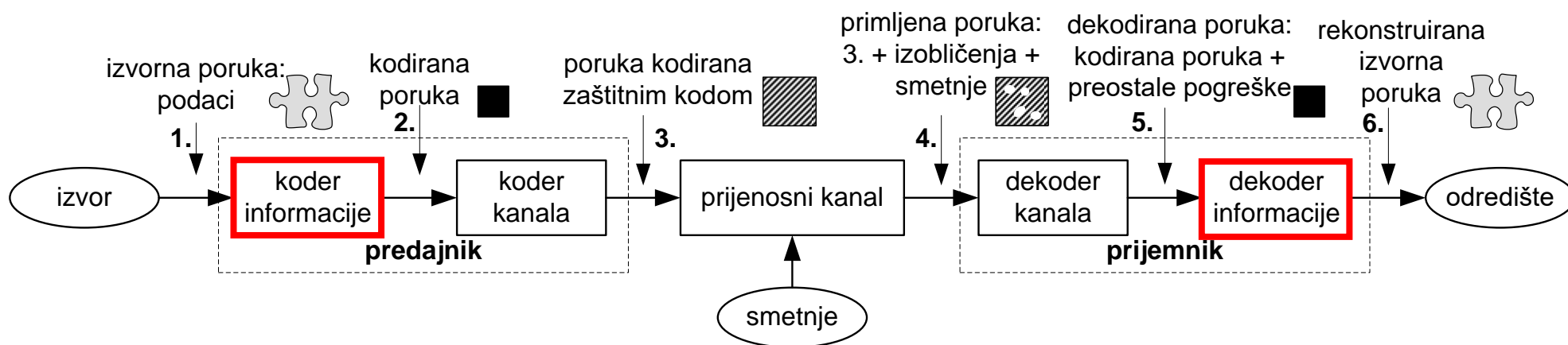


- ◆ Kombinacija izvornog i entropijskog kodiranja
- ◆ Primjene na razne vrste medija (slika, video, glazba...)
- ◆ Različite metode, uglavnom s gubicima (ljudima manje ili više neprimjetnima)
- ◆ Omjer kompresije ovisan o sadržaju i (subjektivnoj) kvaliteti

- ◆ Zvuk
 - PCM, ADPCM, MPEG audio,...
- ◆ Nepomična slika
 - GIF, JPEG, JPEG 2000,...
- ◆ Video
 - H.261, MPEG video...
- ◆ 3D modeli i animacija
 - MPEG-4

Odabrane metode hibridnog kodiranja zvuka i slike obrađene su u predavanjima koja slijede.

Kompresija u sklopu komunikacijskog sustava



Entropijsko kodiranje

- ♦ Uvod u entropijsko kodiranje
- ♦ Karakteristike izvora informacije
 - Stacionarni izvor, ergodički izvor, izvori s memorijom (Markovljevi)
- ♦ Vrste kodova i njihova svojstva
 - Singularni, nesusingularni, jednoznačno dekodabilni, prefiksni kodovi
- ♦ Optimalno kodiranje
- ♦ Metode entropijskog kodiranja
 - Huffmanovo kodiranje
 - Aritmetičko kodiranje
 - Metode rječnika (LZ77, LZ78, LZW)
 - Metode skraćivanja niza (potiskivanje nula, slijedno kodiranje)

- ◆ Osnovna ideja: skraćeno zapisati višestruko ili često ponavljane simbole ili nizove simbola
- ◆ Zajedničko svim metodama entropijskog kodiranja:
 - temelje se direktno na teoriji informacije
 - kodiranje bez gubitaka
 - omjer kompresije ovisi samo o statističkim svojstvima izvora informacije
 - poruka se promatra isključivo kao niz niz slučajnih vrijednosti, ne uzimaju se u obzir svojstva medija (za razliku od izvornog kodiranja)

- ♦ Izvor informacije promatramo kao stohastički proces, tj. niz slučajnih varijabli:

$$X_1, X_2, \dots, X_n$$

- ♦ Izvor u potpunosti opisan raspodjelom združenih vjerojatnosti pojavljivanja varijabli:

$$P\{(X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\} = p(x_1, x_2, \dots, x_n)$$

- ♦ Općenito, moguća zavisnost među varijablama

Stacionarni izvor

Zavod za telekomunikacije

- Statistička svojstva se ne mijenjaju s vremenom

$$P\{(X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\} = P\{(X_{1+l}, X_{2+l}, \dots, X_{n+l}) = (x_1, x_2, \dots, x_n)\},$$

$$\forall l, (x_1, x_2, \dots, x_n) \in X^n, n > 0$$

- Trivijalan primjer stacionarnog izvora:

AEAEAEAEAEAEAE.....

- Trivijalan primjer nestacionarnog izvora:

AEAAEEAAAEAAAEAAAEAAAEAAAEAAAE...

Ergodički izvor

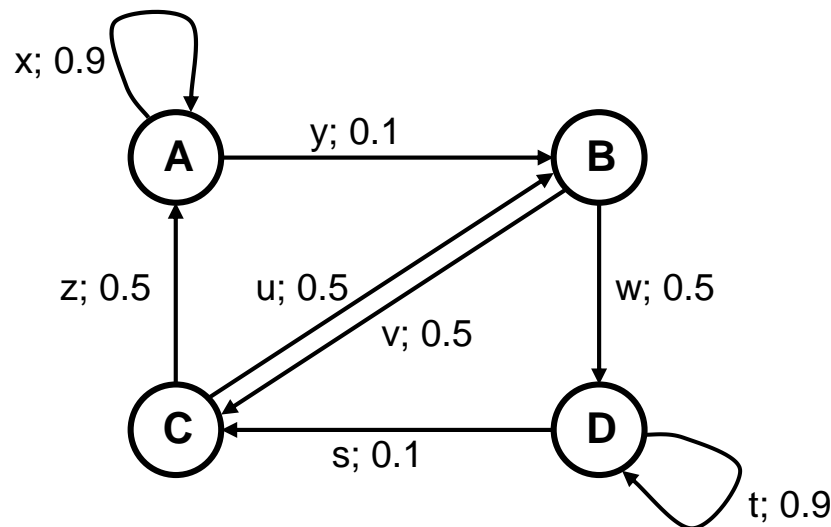
- ◆ Izvor kao skup svih mogućih proizvedenih nizova
 - Prosjek po skupu: prosjek pojavljivanja simbola na nekom mjestu u nizu, gledano među svim nizovima
 - Prosjek po vremenu: učestalost pojavljivanja simbola unutar pojedinog niza
- ◆ Ergodičnost: prosjek po skupu = prosjek po vremenu
- ◆ Svaki proizvedeni niz ima ista svojstva i ona se ne mijenjaju u vremenu
- ◆ Za entropijsko kodiranje promatramo ergodičke izvore (aproksimacija stvarnih izvora)

- ♦ Izvor počinje $\frac{1}{3}$ sa A, $\frac{1}{3}$ B i $\frac{1}{3}$ E
 - Ako počne sa A ili B ponavlja ih izmjenično
 - Ako počne sa E, ponavlja samo E
 - Skup mogućih nizova:
 - Niz 1: ABABABABABABAB...
 - Niz 2: BABABABABABABA...
 - Niz 3: EEEEEEEEEEEEEEEE...

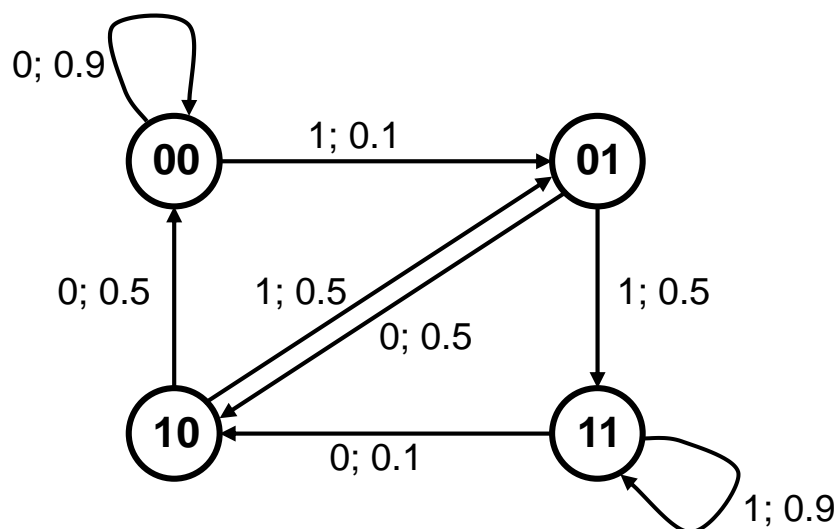
Simbol	Prosjek po vremenu za niz 1	Prosjek po vremenu za niz 2	Prosjek po vremenu za niz 3	Prosjek po skupu
A	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{3}$
B	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{3}$
E	0	0	1	$\frac{1}{3}$

- ◆ Vjerojatnost pojavljivanja simbola je ovisna o jednom ili više prethodnih simbola
- ◆ Neki nizovi simbola vjerojatniji od drugih
- ◆ Većina prirodnih izvora su izvori s memorijom
 - Npr. iz slova u tekstu, zvuk govora, slika

- ♦ Izvori s memorijom često se mogu opisati pomoću Markovljevih
- ♦ Stanja, vjerojatnosti prijelaza
- ♦ Pri prijelazu stanja generira se simbol



- ♦ Binarni Markovljev izvor s memorijom od dva simbola



- ♦ Tipičan izlaz:

0000000000000000001111111111111111110000111111111111000001111111111111....

Kodiranje

Zavod za telekomunikacije

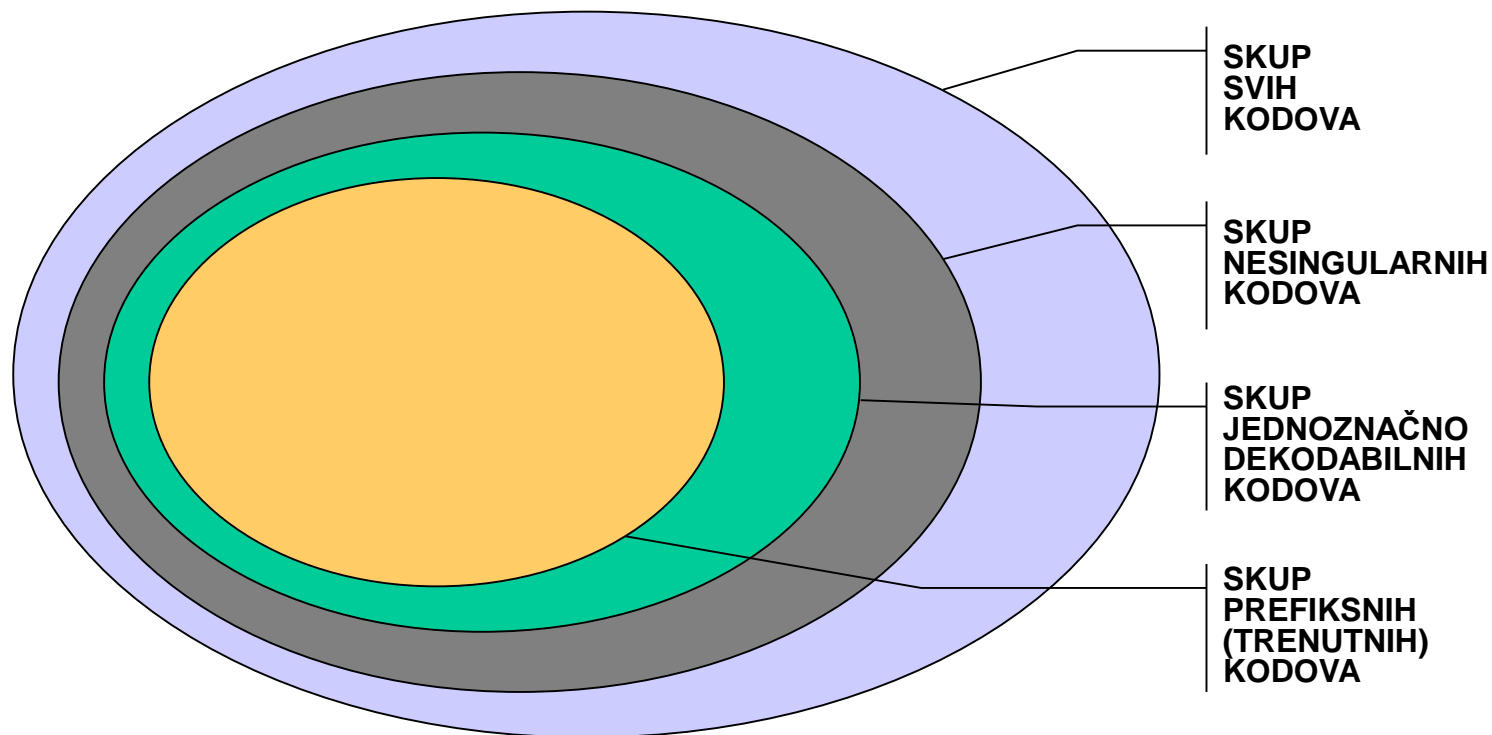
- ◆ Dodjela kodnih riječi simbolima poruke

$$X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$$

$$x_i \in X \xrightarrow{\text{KODIRANJE}} C(x_i)$$

$$C(x_i) \in D^*, D = \{a_1, a_2, \dots, a_d\},$$

- ◆ Kodiranje sa svojstvom sažimanja: kompresija
- ◆ U praksi gotovo uvijek binarna abeceda
 - $d = 2, D = \{0, 1\}$
 - Izlaz koder: struja bitova (engl. *bitstream*)



- ♦ Minimum prosječne duljine kodne riječi se dobiva za:

$$l_i^* = -\log_d p_i \Rightarrow L = -\sum_{i=1}^n p_i \log_d p_i = H(X)$$

- ♦ Ali l_i moraju biti cijeli brojevi, pa se ne može uvijek postići $L=H$:

$$L \geq H(X)$$

- ♦ Za optimalni kod, prosječna duljina kodne riječi je unutar jednog bita od entropije: $H(X) \leq L < H(X) + 1$
- ♦ Efikasnost koda: $\varepsilon = \frac{H(X)}{L}$

- ◆ **Huffmanovo kodiranje**
 - optimalno kodiranje
 - binarno stablo
 - kraći zapis čestih znakova
- ◆ **Aritmetičko kodiranje**
 - poopćenje Huffmanovog kodiranja
 - cijela poruka se pretvara u jednu kodnu riječ
- ◆ **Metode rječnika**
 - isti rječnik kodnih riječi na strani pošiljatelja i primatelja
 - dinamička konstrukcija rječnika
 - Lempel-Ziv (LZ77, LZ78), Lempel-Ziv-Welch (LZW)
- ◆ **Metode skraćivanja niza**
 - potiskivanje nula, slijedno kodiranje

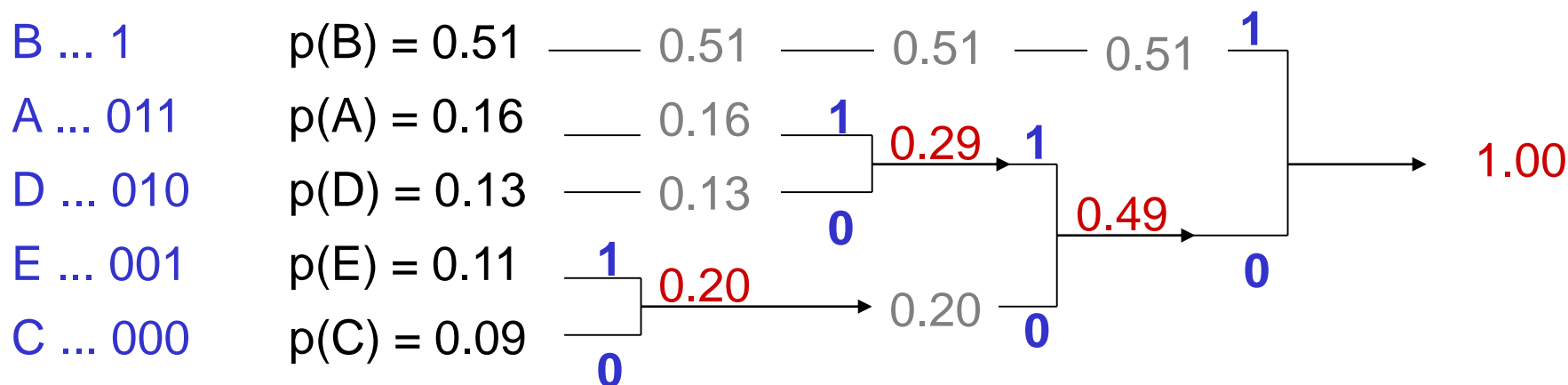
- ◆ D. A. Huffman, 1952. godine
- ◆ Kodira pojedinačne simbole kodnim riječima promjenjive duljine, ovisno o (poznatim!) vjerojatnostima njihova pojavljivanja
- ◆ Temelji se na dvije jednostavne činjenice:
 - (1) U optimalnom kodu, simboli s većom vjerojatnošću pojavljivanja imaju kraće kodne riječi od onih s manjom vjerojatnošću
 - (2) U optimalnom kodu, dva simbola s najmanjim vjerojatnostima imaju kodne riječi jednake duljine (vrijedi za prefiksni kod)
- ◆ Ishod: sažetiji zapis (npr. tipičan tekst se sažima za 45%)

- ◆ Algoritam stvaranja koda:
 1. Sortiraj simbole po padajućim vjerojatnostima
 2. Pronađi dva simbola s najmanjim vjerojatnostima
 3. Jednom od njih dodijeli simbol “0”, drugom “1”
 4. Kombiniraj ta dva simbola u jedan nadsimbol (nadsimbol je novi simbol čija je vjerojatnost pojavljivanja jednaka zbroju vjerojatnosti pojavljivanja dvaju simbola od kojih je nastao) i zapiši ih kao dvije grane binarnog stabla, a nadsimbol kao račvanje iznad njih
 5. Ponavljaj 1-4 dok ne dobiješ samo jedan nadsimbol
 6. Povratkom kroz stablo očitaj kodove
- ◆ Podatkovna struktura algoritma je binarno stablo
- ◆ Algoritam dekodiranja koristi isti postupak za gradnju stabla
 - Dekoder mora znati vjerojatnosti pojavljivanja simbola

Huffmanovo kodiranje: primjer

Zavod za telekomunikacije

- Skup simbola {A, B, C, D, E} s vjerojatnostima pojavljivanja $p(A) = 0.16$, $p(B) = 0.51$, $p(C) = 0.09$, $p(D) = 0.13$, $p(E) = 0.11$
- Za uniformni kod, prosječna duljina koda je **3 bit/simbol** (jer je $2^2 \leq 5 \leq 2^3$).
- Entropija: **1.96 bit/simbol**



- Prosječna duljina dobivenog koda u našem slučaju je:

$$L = \sum_{x \in X} p_x l_x = 3 \times (0.09 + 0.11 + 0.13 + 0.16) + 0.51 = 1.98 \text{ bit/simbol}$$

- ◆ kodiranje je idealno ako su vjerojatnosti $1/2, 1/4, \dots, 1/2^n$
- ◆ u stvarnim slučajevima to obično nije slučaj, te rezultat ovisi o vjerojatnostima pojavljivanja simbola
- ◆ prednosti:
 - jednostavan za izvedbu
 - vrlo dobro kodiranje za „dobre“ vjerojatnosti pojavljivanja simbola
- ◆ nedostaci:
 - vjerojatnosti pojavljivanja simbola moraju biti poznate; ovise o primjeni (tekst, slika)
 - za “loše raspoređene” vjerojatnosti pojavljivanja dobiju se izrazito loši kodovi

Primjer lošeg koda i prošireni Huffmanov kod

Simbol	Vjerojatnost	Kodna riječ
a_1	0.95	0
a_2	0.02	10
a_3	0.03	11

PROŠIRENI KOD		
Simbol	Vjerojatnost	Kodna riječ
a_1a_1	0.9025	0
a_1a_2	0.0190	111
a_1a_3	0.0285	100
a_2a_1	0.0190	1101
a_2a_2	0.0004	110011
a_2a_3	0.0006	110001
a_3a_1	0.0285	101
a_3a_2	0.0006	110010
a_3a_3	0.0009	110000

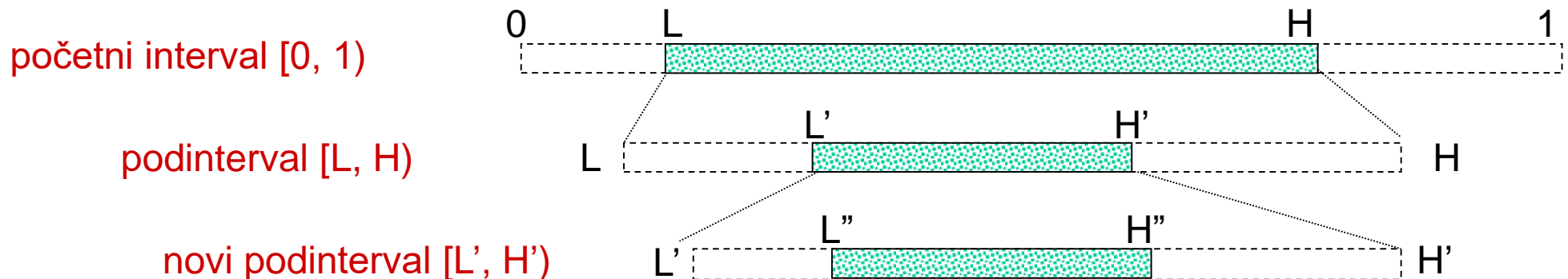
- ♦ Entropija: 0.335 bit/simbol
- ♦ Prosječna duljina: 1.05 bit/simbol: **213% više od entropije!!**
- ♦ Prošireni kod: $1.222 / 2 = 0.611$ bit/simbol: 72% više od entropije.
- ♦ Bolje je kodirati duže sekvence, ali tada broj kodnih riječi raste eksponencijalno

- ◆ Česta primjena unutar složenijih algoritama
- ◆ Primjeri:
 - standardi za telefaks (T.4, T.6)
 - standard za nepomičnu sliku JPEG

- ♦ Autori Pasco & Rissanen (nezavisno), 1976. godine
- ♦ Algoritam uzima kao ulaz cijele nizove simbola (“poruke”) i preslikava ih na realne brojeve, ovisno o (poznatim!) statističkim svojstvima

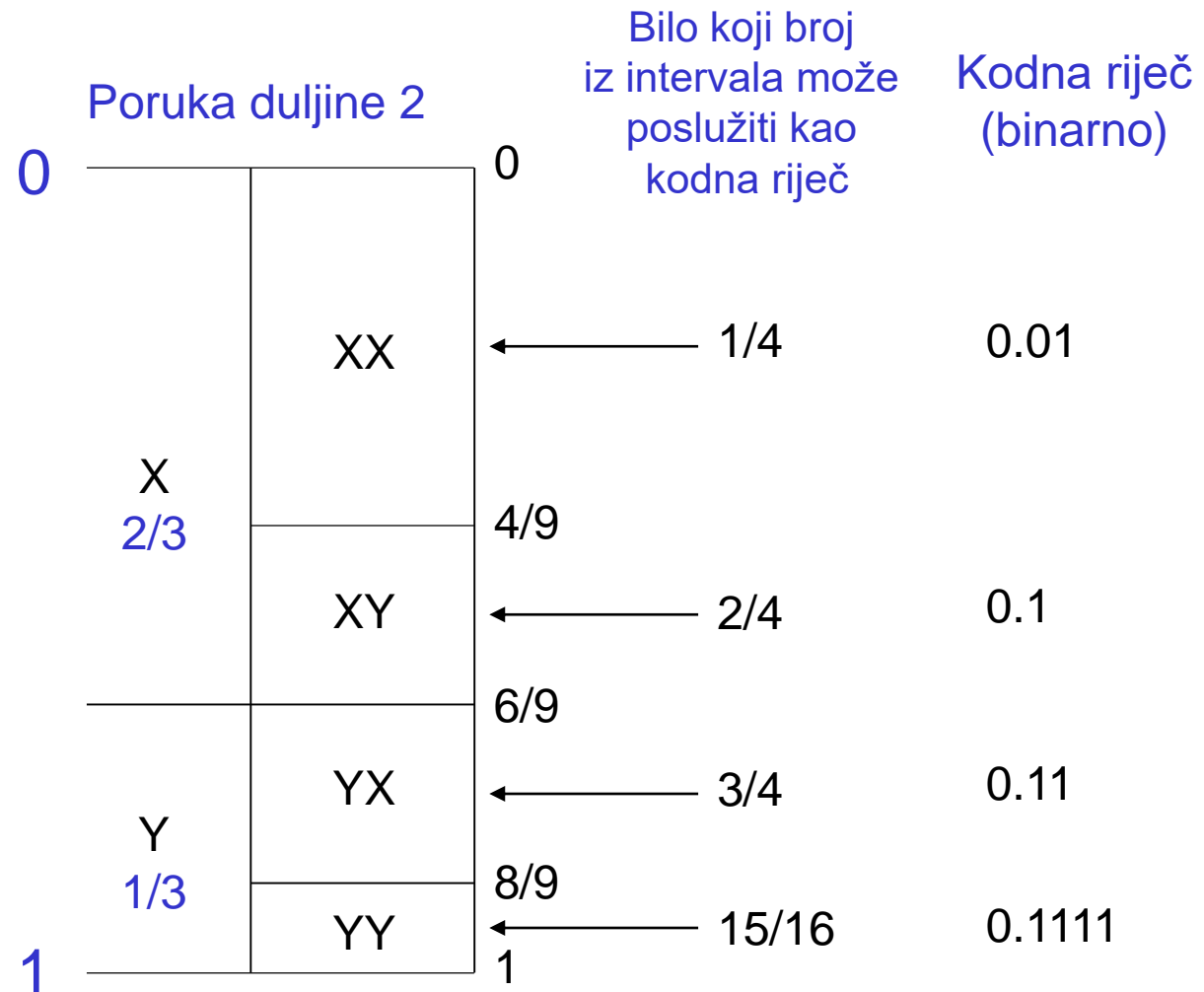
Aritmetičko kodiranje: postupak

1. Podijeli interval $[0, 1)$ u n podintervala koji odgovaraju simbolima iz abecede; duljina svakog podintervala proporcionalna vjerojatnosti odgovarajućeg simbola
2. Iz promatranog skupa podintervala, odaberi podinterval koji odgovara sljedećem simbolu u poruci
3. Podijeli taj podinterval u n novih podintervala, proporcionalno vjerojatnostima pojavljivanja simbola iz abecede; tako nastaje novi skup podintervala koji promatramo
4. Ponavlja korake 2 i 3 dok cijela poruka nije kodirana
5. Konačni kod za čitavu poruku je jedan broj iz intervala u binarnom obliku



Aritmetičko kodiranje: primjer (1)

- $M=2$
- simboli: X, Y
 $p(X) = 2/3$
 $p(Y) = 1/3$
- poruka duljine 2
(moguće poruke
XX, XY, YX, YY)
kodira se onim
brojem bita
dovoljnim za
jedinstveno
određivanje
intervala
(binarni razlomak!)



Aritmetičko kodiranje: primjer (2)

- primjer za poruku duljine 3

- $M=2$

- simboli:

X, Y

$$p(X) = 2/3$$

$$p(Y) = 1/3$$

Kodna riječ
(binarno)

X	XX	XXX	←	1/4	0.01
		XXY	←	3/8	0.11
	XY	XYX	←	4/8	0.100
		XYY	←	10/16	0.1010
	Y	YX	YXX	←	6/8
YXY			←	14/16	0.1110
YY		YYX	←	15/16	0.1111
		YYY	←	31/31	0.11111

1. Podijeli početni interval $[0, 1)$ u podintervale po vjerojatnostima pojavljivanja simbola
2. Uzmi primljeni kod kao realni broj
3. Pronađi podinterval u kojem se nalazi broj (kod)
4. Zapiši simbol koji odgovara tom podintervalu
5. Podijeli taj podinterval u n novih podintervala, proporcionalno vjerojatnostima pojavljivanja simbola iz abecede; tako nastaje novi skup podintervala koji promatramo
6. Ponavljaj korake 3-5 dok ne dođe kraj poruke

Dekodiranje: primjer

- primjer za poruku
duljine 3

- $M=2$

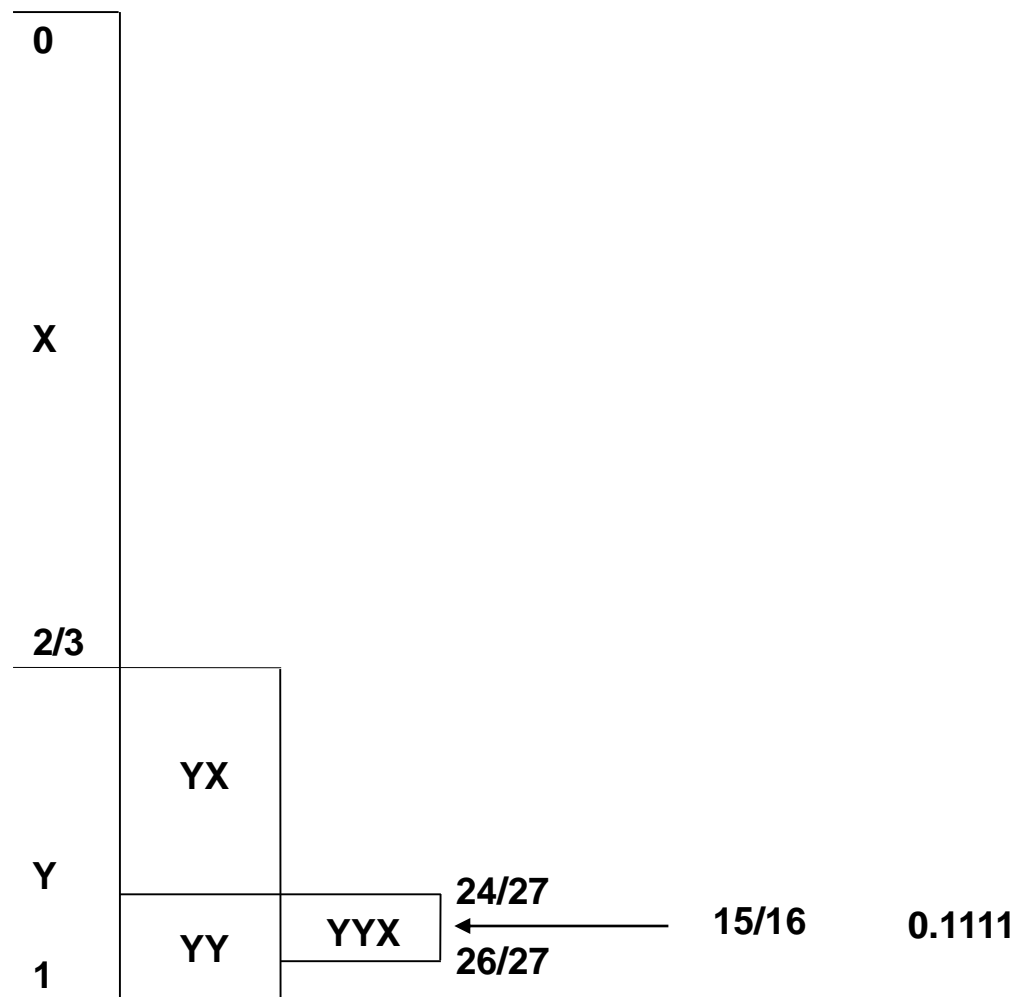
- simboli:

X, Y

$$p(X) = 2/3$$

$$p(Y) = 1/3$$

- Primljeni kod 1111
tj. 15/16



- ♦ Kojim brojem iz podintervala kodirati poruku?
- ♦ Može se uzeti bilo koja vrijednost iz podintervala
- ♦ Dovoljan broj znamenki:

$$l(x) = \left\lceil \log \frac{1}{P(x)} \right\rceil + 1 \text{ [bit]}$$

- ♦ Na ovakav način dobiva se uvijek prefiksni kod

- ◆ Do sada opisani algoritam neupotrebljiv
 - Neprihvatljivo čekanje do kraja poruke
 - Algoritam podrazumijeva beskonačnu preciznost realnih brojeva – na računalu prikaz s pomičnim zarezom
 - Operacije s realnim brojevima su skupe
- ◆ Potreban je algoritam koji:
 - Koristi operacije sa cijelim brojevima
 - Koristi prikaz sa fiksnim brojem bitova
 - Proizvodi simbole koda tokom postupka kodiranja, a ne na kraju

- ◆ Osnovni postupak podjele na podintervale je isti
- ◆ Koristi se fiksni broj znamenki za prikaz intervala
- ◆ Kada je prva znamenka u prikazu gornje i donje granice ista, interval se *renormalizira*:
 - Prvih n znamenki se šalje na izlaz koder
 - Znamenke se pomiču ulijevo za jedno mjesto
 - Desno se dodaje znamenka: 0 na donju, 1 na gornju granicu intervala (ako su znamenke binarne)

Renormalizacija: primjer

x	p(x)
RAZMAK	1/10
A	1/10
B	1/10
E	1/10
G	1/10
I	1/10
L	2/10
S	1/10
T	1/10

	GORNJA GRANICA	DONJA GRANICA	DULJINA INTERVALA	KUMULATIVNI IZLAZ
Početno stanje	99999	00000	100000	
Kodiraj B (0.2-0.3)	29999	20000		
Renormalizacija, izlaz: 2	99999	00000	100000	.2
Kodiraj I (0.5-0.6)	59999	50000		.2
Renormalizacija, izlaz: 5	99999	00000	100000	.25
Kodiraj L (0.6-0.8)	79999	60000	20000	.25
Kodiraj L (0.6-0.8)	75999	72000		.25
Renormalizacija, izlaz: 7	59999	20000	40000	.257
Kodiraj RAZMAK (0.0-0.1)	23999	20000		.257
Renormalizacija, izlaz: 2	39999	00000	40000	.2572
Kodiraj G (0.4-0.5)	19999	16000		.2572
Renormalizacija, izlaz: 1	99999	60000	40000	.25721
Kodiraj A (0.1-0.2)	67999	64000		.25721
Renormalizacija, izlaz: 6	79999	40000	40000	.257216
Kodiraj T (0.9-1.0)	79999	76000		.257216
Renormalizacija, izlaz: 7	99999	60000	40000	.2572167
Kodiraj E (0.3-0.4)	75999	72000		.2572167
Renormalizacija, izlaz: 7	59999	20000	40000	.25721677
Kodiraj S (0.8-0.9)	55999	52000		.25721677
Renormalizacija, izlaz: 5	59999	20000		.257216775
Renormalizacija, izlaz: 2				.2572167752
Renormalizacija, izlaz: 0				.25721677520

Usporedba aritmetičko - Huffman



Zavod za telekomunikacije

Huffman	Aritmetičko kodiranje
Kodira svaki simbol posebno	Kodira cijelu poruku jednim kodom: realni broj 0 - 1
Minimalno 1 bit/simbol	Moguće < 1 bit/simbol
Duljina poruke nije važna	Teoretski optimalno za dugačke poruke
Kodiranje niza simbola moguće samo proširenim Huffman kodom	Uvijek se kodira cijela poruka
Jednostavno za računanje	Zahtjevnije za računanje

- ◆ Primjena kao komponente u raznim standardima i za razne vrste medija
- ◆ Dokumenti
 - JBIG (Joint Bi-level Image Processing Group)
- ◆ Slika
 - JPEG
- ◆ Sintetički sadržaji/animacija
 - MPEG-4 FBA (Face and Body Animation)

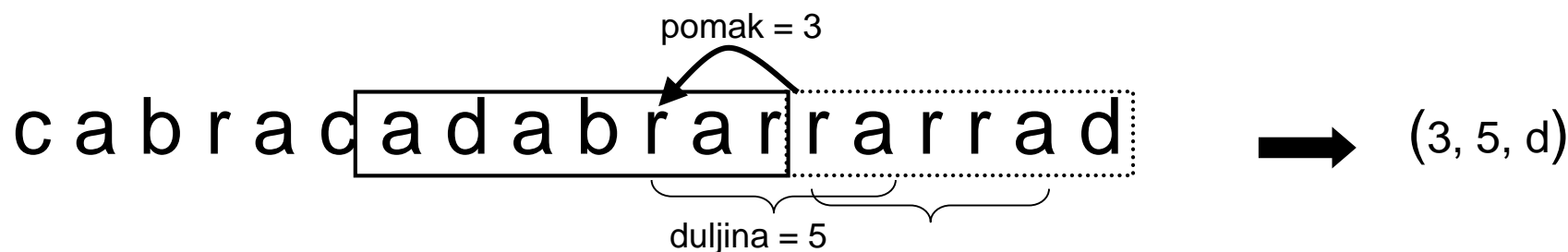
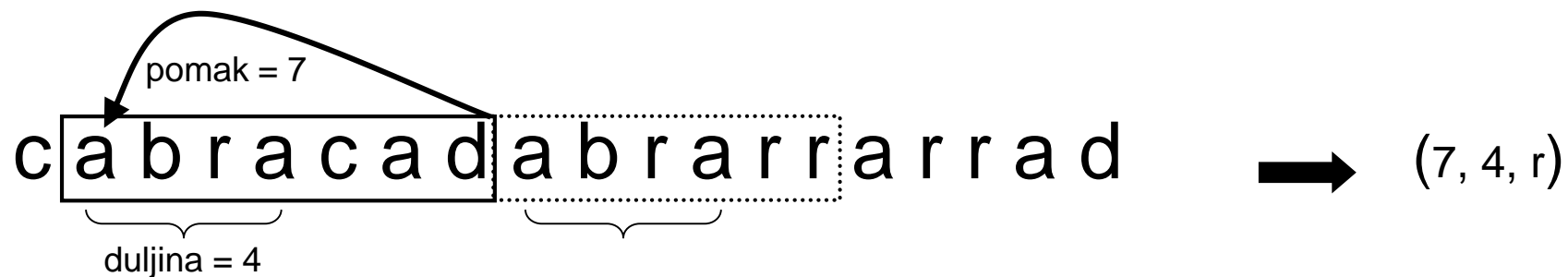
Metode rječnika

- ♦ Algoritmi kodiranja metodama rječnika uzimaju kao ulaz nizove simbola (“riječi”) promjenjive duljine i kodiraju ih kodnim riječima stalne duljine iz rječnika
- ♦ Ne trebaju znati vjerojatnosti pojavljivanja simbola, nazivaju se i *univerzalni koderi*
- ♦ Koder i dekodeer moraju imati isti rječnik
- ♦ Rječnik može biti statičan, no najčešće je prilagodljiv

- ◆ Koder i dekoder dinamički grade rječnik
 - LZ77: Rječnik je posmični prozor
 - LZ78: riječi se grade dodavanjem slova na postojeće riječi (u početku rječnik je prazan)
 - Lempel-Ziv-Welch (LZW) algoritam
 - izvorni algoritam smislili Ziv i Lempel (1977 - LZ77, 1978 - LZ78), a Welch ga je doradio i poboljšao 1984 (zato **LZW**)
 - algoritam relativno jednostavan, iako složeniji od Huffmanovog
 - izvorni LZW algoritam koristi rječnik s 4K riječi, s tim da su prvih 256 riječi standardni ASCII kodovi

- ♦ Rječnik je posmični prozor od N zadnjih simbola
- ♦ U svakom koraku traži se u rječniku najduži niz simbola jednak nadolazećim simbolima, te se kodira kao uređena trojka (*pomak*, *duljina*, *sljedeći_simbol*)
- ♦ Nedostatak: “kratka” memorija

LZ77: primjer kodiranja



- ◆ Umjesto posmičnog prozora, zasebna memorija za rječnik
 - Rječnik je poredana lista riječi (nizova simbola)
 - Riječ se dovaća pomoću indeksa (rednog broja)
- ◆ LZ78
 - Rječnik u početku prazan
 - U svakom koraku šalje se (*indeks, idući simbol*)
 - Indeks pokazuje na najdulju riječ u rječniku jednaku nadolazećem nizu simbola
 - Rječnik se nadopunjava novim riječima tijekom kodiranja

- Algoritam kodiranja:

```
1. RadnaRiječ = slijedeći simbol sa ulaza
2. WHILE (ima još simbola na ulazu) DO
3.     NoviSimbol = slijedeći simbol sa ulaza
4.     IF RadnaRiječ+NoviSimbol postoji u rječniku THEN
5.         RadnaRiječ = RadnaRiječ+NoviSimbol
6.     ELSE
7.         IZLAZ: kod za RadnaRiječ
8.         dodaj RadnaRiječ+NoviSimbol u rječnik
9.         RadnaRiječ = NoviSimbol
10.    END IF
11. END WHILE
12. IZLAZ: kod za RadnaRiječ
```


Kodiranje algoritmom LZW: primjer

Sadržaj rječnika na početku:

kodna riječ	znak
(1)	A
(2)	B
(3)	C

Niz znakova koje treba kodirati:

Mjesto	1	2	3	4	5	6	7	8	9
Simbol	A	B	B	A	B	A	B	A	C

LZW:

korak	mjesto	sadržaj rječnika	izlaz iz kodera
1.	1	(4) A B	(1)
2.	2	(5) B B	(2)
3.	3	(6) B A	(2)
4.	4	(7) A B A	(4)
5.	6	(8) A B A C	(7)
6.	9		(3)

LZW kodiranje: primjer dekodiranja

KORAK	RADNA RIJEČ	ULAZ DEKODERA	DEKODIRANI SIMBOLI	SADRŽAJ RJEČNIKA
1		(1)	A	
2	A	(2)	B	(4) AB
3	B	(2)	B	(5) BB
4	B	(4)	AB	(6) BA
5	AB	(7)	ABA	(7) ABA
6	ABA	(3)	C	

- ♦ LZW
 - UNIX compress
 - GIF
 - Modem V.24 bis
- ♦ LZ77
 - ZIP

- ♦ primjer - potiskivanje nula:

89400

- ◆ **slijedno kodiranje** (engl. *run-length encoding*)
broj ponavljanja
- ◆ algoritam kodiranja temelji se na kraćem zapisu ponavljanih simbola pomoću specijalnog znaka (!)
- ◆ primjer: ABCCCCCCCCCDEFFFEABC

ABCCCCCCCC

8 okteta

DEFFFABC...

3 okteta

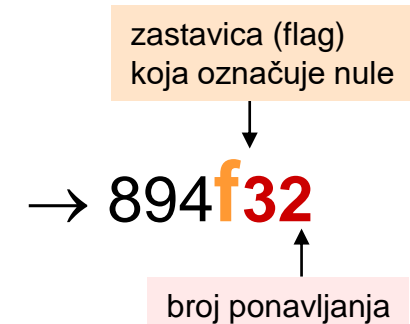
ABC!8

3 okteta

DEFFFABC...

3 okteta

← “isplati” se za 4+ znakova



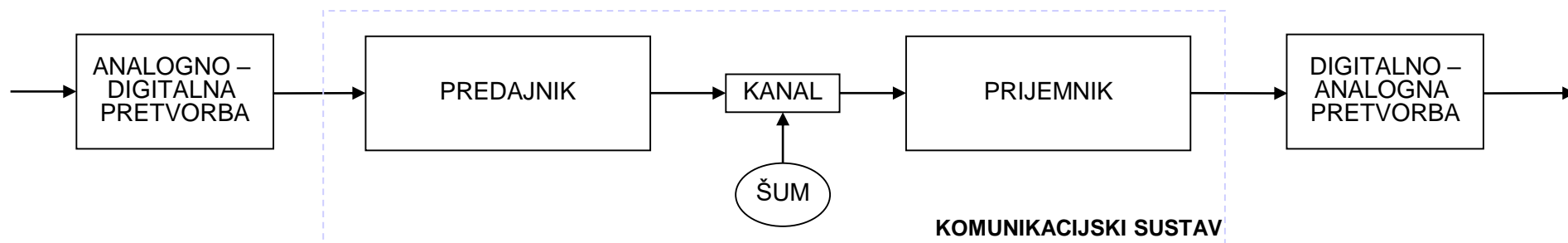
- ◆ Primjena: prva generacija telefaksa, unutar JPEG-a

Izvorno kodiranje

- ◆ Uvod
 - Svojstva metoda izvornog kodiranja
 - Analogni mediji u diskretnom kom. sustavu
 - Principi kompresije pri izvornom kodiranju
- ◆ Osnovne metode izvornog kodiranja
 - Kvantizacija
 - Poduzorkovanje
 - Transformacijsko kodiranje
 - Diferencijalno (predikcijsko) kodiranje
 - Potpojasno kodiranje
 - Kodiranje zasnovano na modelu

- ◆ Najčešće sažimaju s gubicima
- ◆ Koriste semantiku izvora, tj. posebna svojstva pojedinih medija
- ◆ Koriste karakteristike ljudske percepcije medija za bolju kompresiju uz malo primjetnu pogrešku
- ◆ Omjer kompresije jako ovisan o sadržaju
- ◆ Koriste se u sklopu hibridnih metoda za kodiranje pojedinih medija
 - Obično prvo izvorno kodiranje, zatim entropijsko

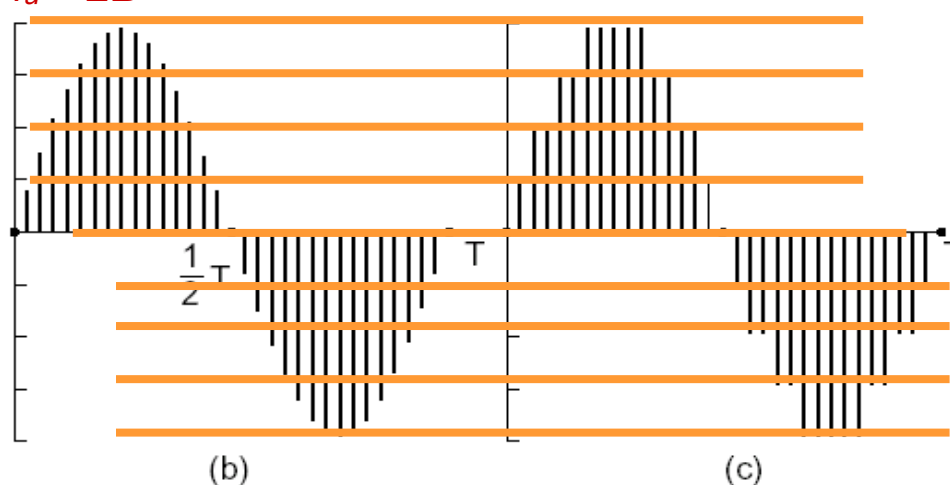
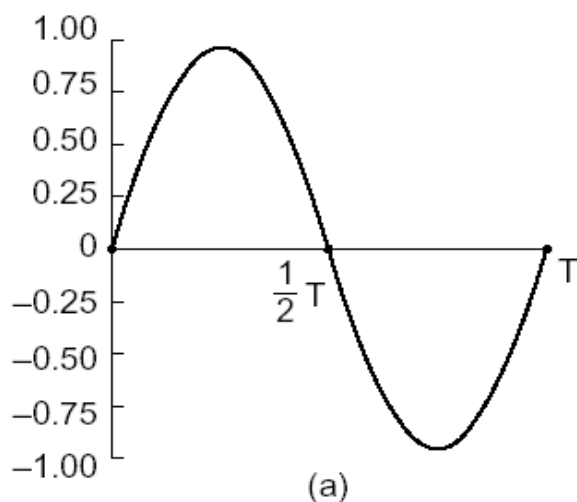
Analogni mediji u diskretnom komunikacijskom sustavu



izvorni signal \longrightarrow uzorkovanje f_u \longrightarrow kvantizacija \longrightarrow *poruka*

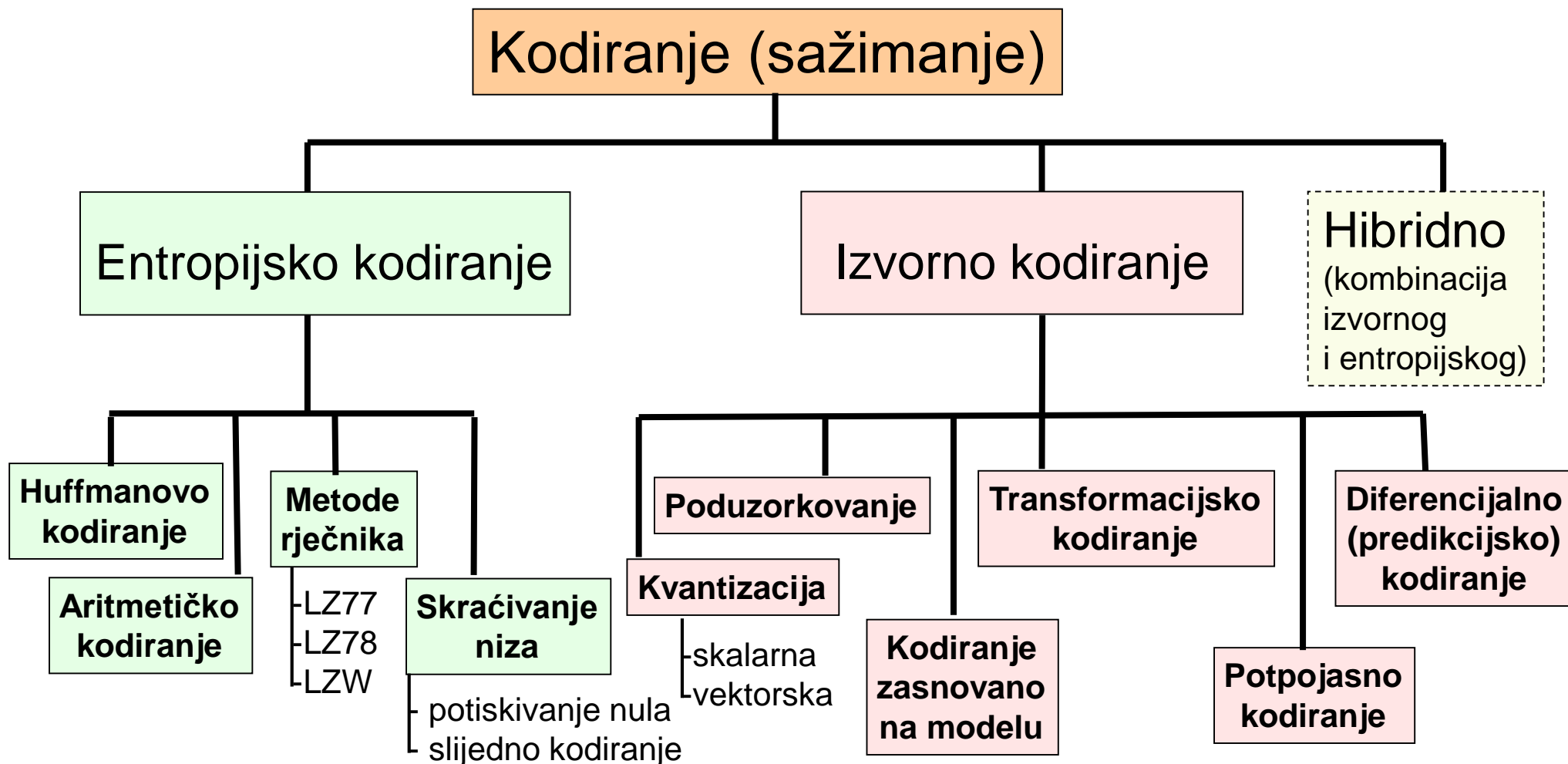
Nyquistov kriterij:

$$f_u \geq 2B$$

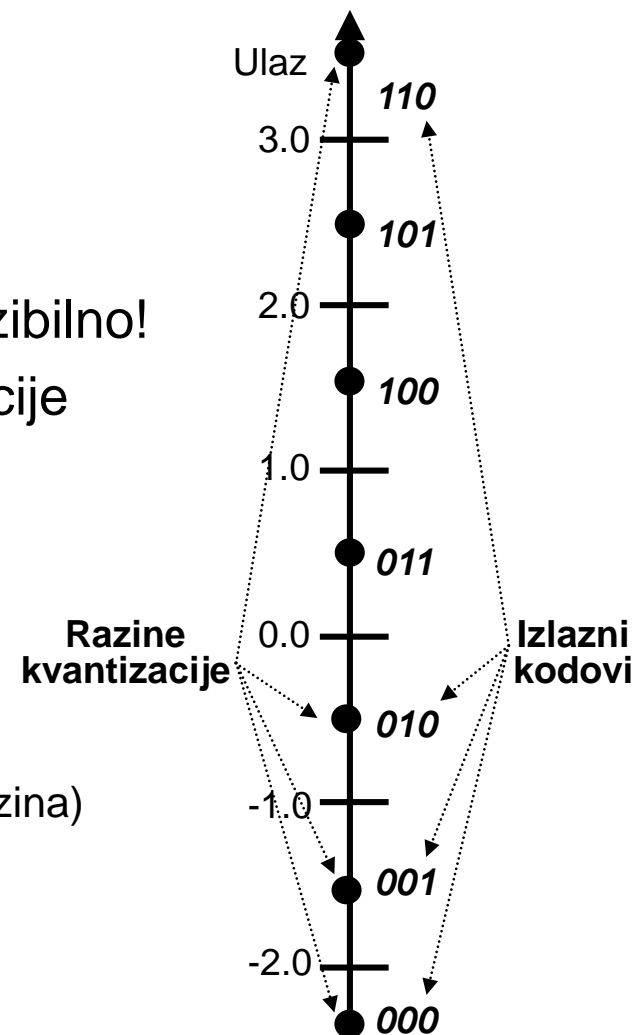


{0, 20, 38, 55, 73...}

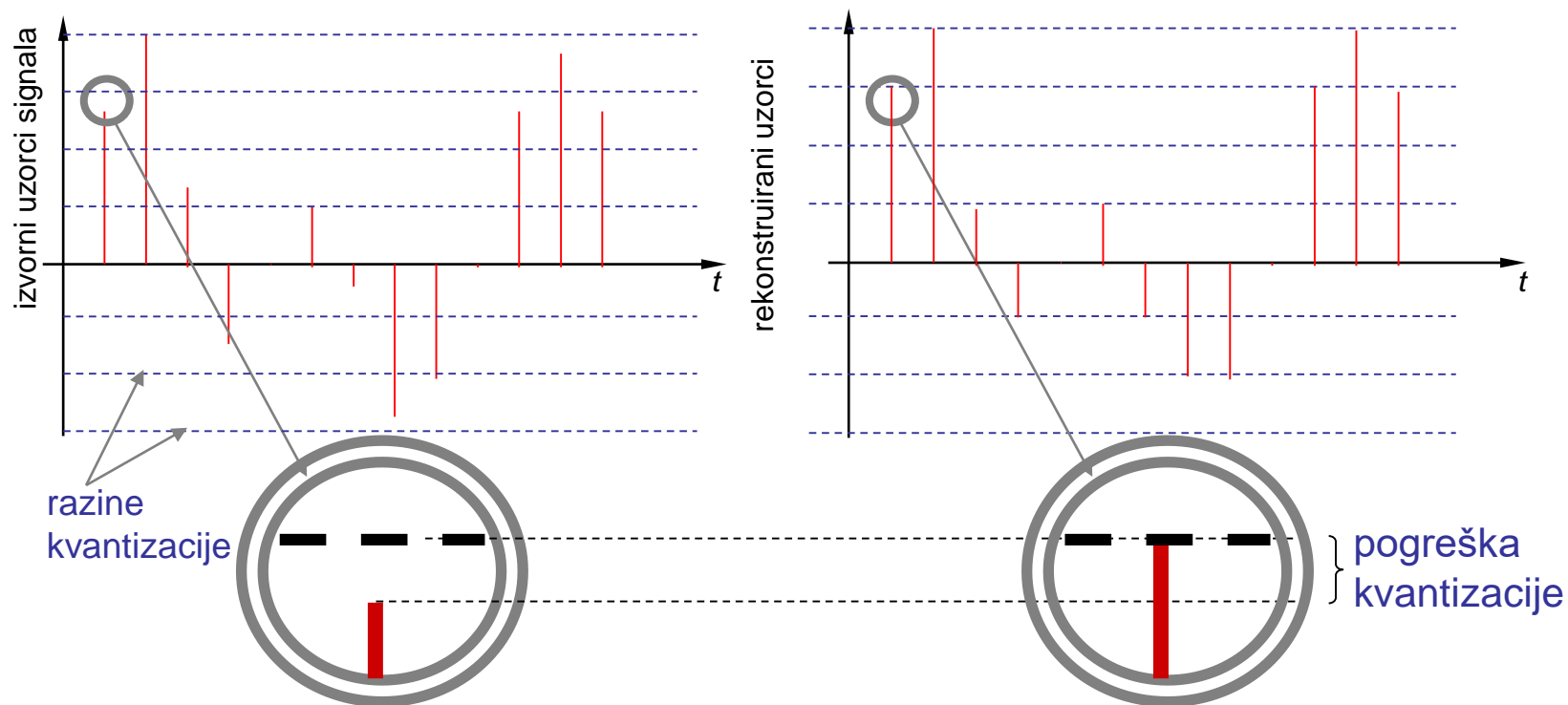
- ◆ Uklanjanje zalihosti (redundancije):
 - Vremenska zalihost (npr. korelacija uzastopnih uzoraka zvučnog signala)
 - Prostorna zalihost (npr. korelacija susjednih elemenata slike - pixela)
 - Spektralna zalihost (npr. korelacija između boja ili svjetline u slici)
- ◆ Uklanjanje irelevantnosti:
 - Granice ljudske precepcije
 - Spuštenje razine kvalitete reprodukcije



- ◆ Aproksimacija signala konačnim skupom kodova
 - Raspon ulaza podijeljen na intervale
 - Sve vrijednosti u intervalu isti kod – nije reverzibilno!
 - Kod intervala dekodira se kao razina kvantizacije
 - Svaka vrijednost se svodi na jednu razinu kvantizacije
- ◆ A/D pretvorba uvijek uvodi kvantizaciju
- ◆ Izuzetno korisna kao metoda kompresije
 - Primjer: zvuk 16 bit (65536 razina) → 8 bit (256 razina)
- ◆ Linearna i nelinearna kvantizacija
- ◆ Skalarna i vektorska kvantizacija

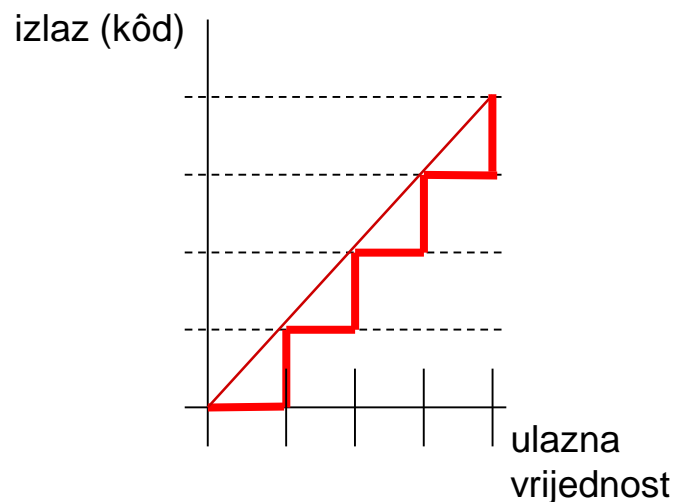


Pogreška kvantizacije

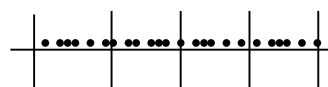


Linearna i nelinearna kvantizacija

a) Linearna kvantizacija



1) Jednolika raspodjela ulaza



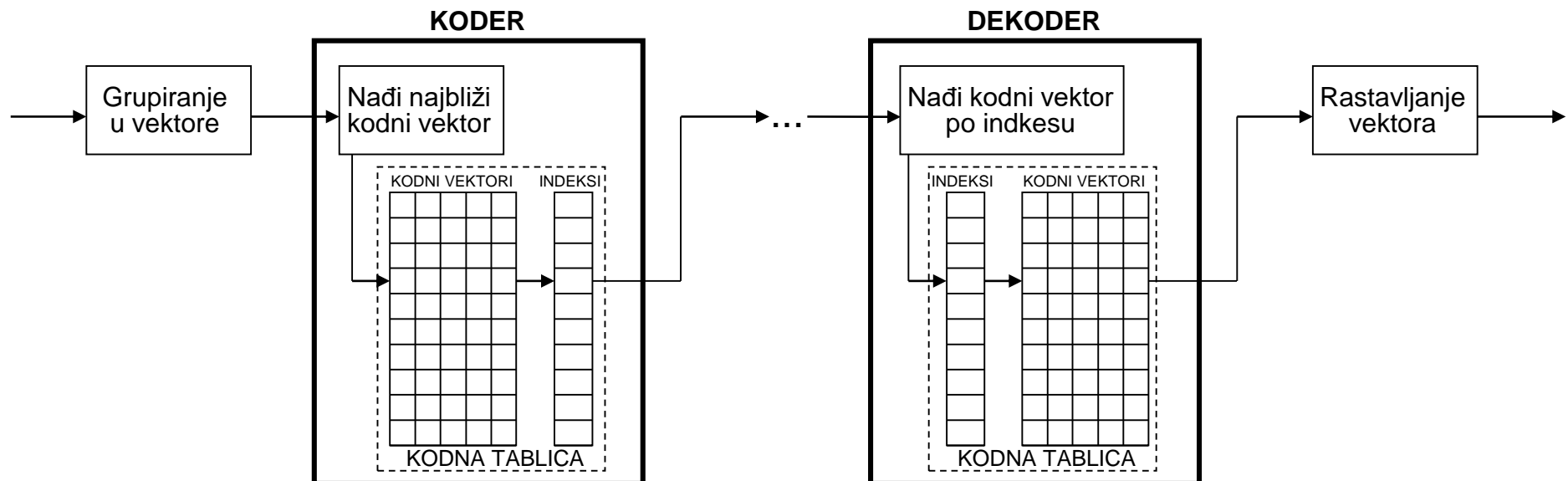
2) Nejednolika raspodjela ulaza



b) Nelinearna kvantizacija



- ◆ Podaci se grupiraju u n -dimenzionalne vektore
- ◆ Za svaki vektor se u *kodnoj tablici* pronalazi najbliži *kodni vektor*; njegov *indeks* je kod
- ◆ Za dekodiranje se koristi ista kodna tablica



Primjer: 2D vektorska kvantizacija

- ◆ Kodna tablica:

- $i = 1:$ $y_1 = (0, 0)$

- $i = 2:$ $y_2 = (2, 1)$

- $i = 3:$ $y_3 = (1, 3)$

- $i = 4:$ $y_4 = (1, 4)$

- ◆ Poruka: 0 1 2 3 2 0

- ◆ Kodirano: 1 3 2

- ◆ Dekodirana poruka:

0 0 1 3 2 1

- ◆ Pogreška kvantizacije:

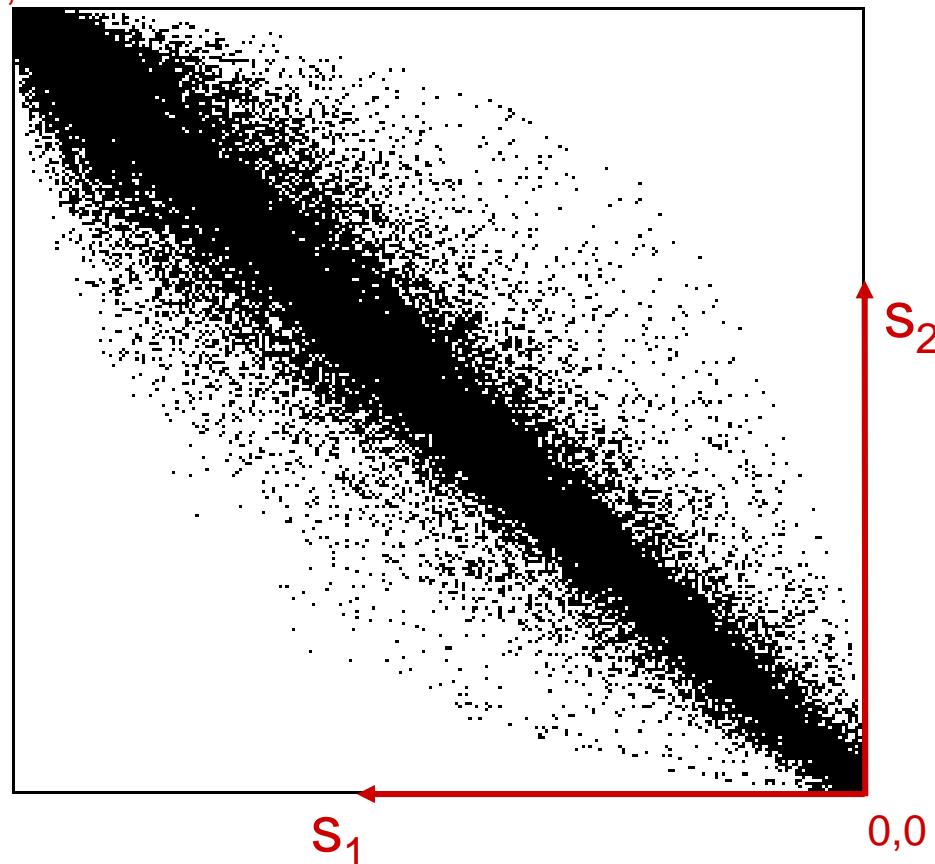
0 -1 -1 0 0 1

Primjer: vektorska kvantizacija slike (1/2)



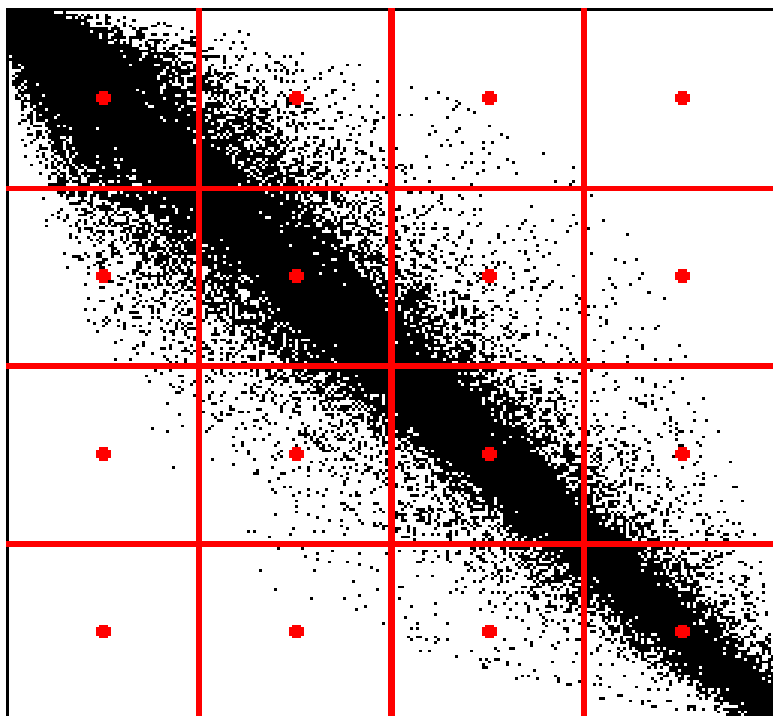
Izvorna slika sastoji se od točaka b_i ; svaka točka je vrijednost 0-255

255,255

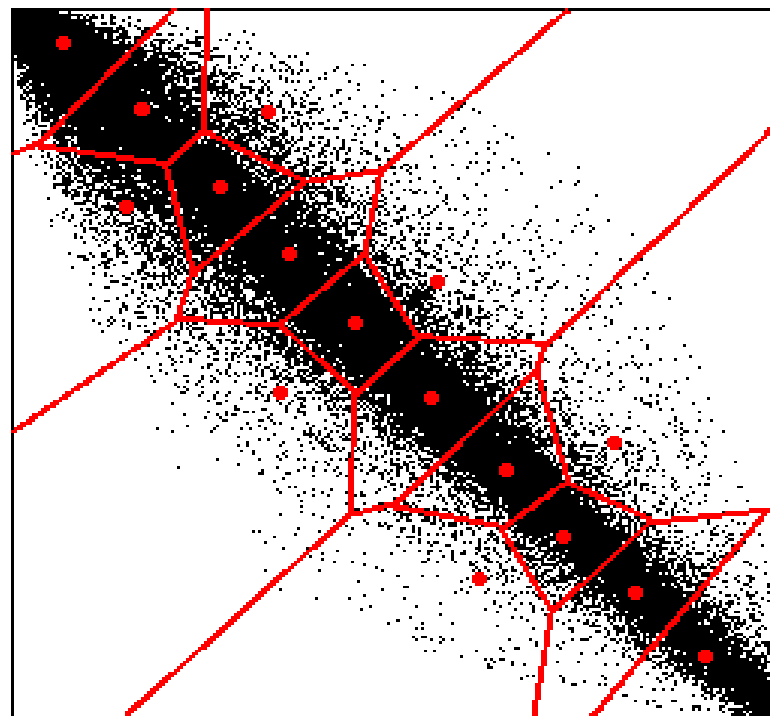


Parovi svjetlina susjednih točaka slike

Primjer: vektorska kvantizacija slike (2/2)



Skalarna kvantizacija
prikazana u obliku
vektorske kvantizacije



Vektorska kvantizacija

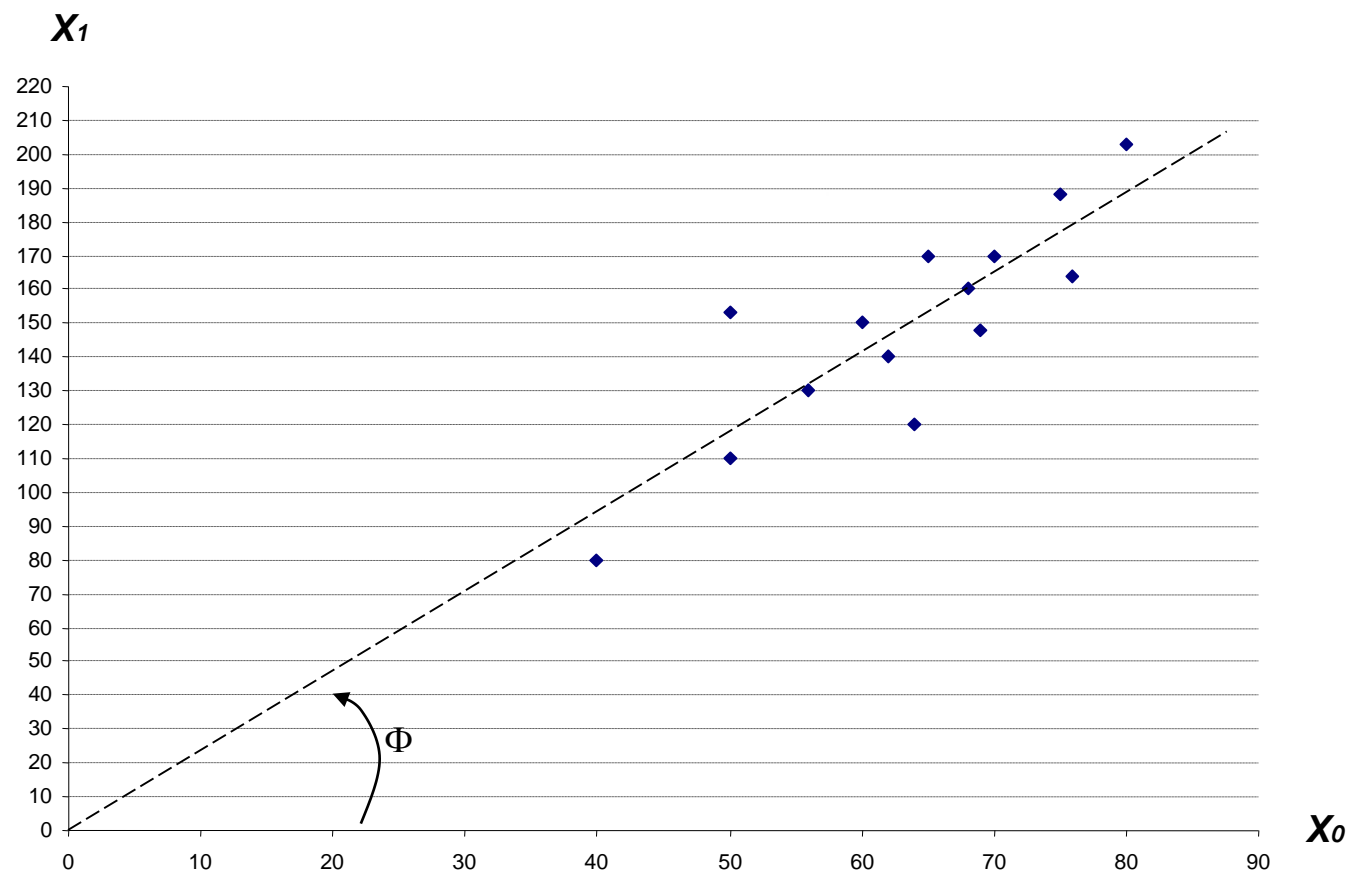
- ◆ Smanjivanje frekvencije uzorkovanja
 - Slika: rezolucija
 - Zvuk: broj uzoraka zvuka u sekundi
- ◆ Smanjivanje broja uzoraka = sažimanje
- ◆ Nyquist: granica poduzorkovanja bez pogreške
- ◆ Svjesno unošenje pogreške
 - Uzima se u obzir:
 - ograničenja ljudske percepcije
 - posebnosti pojedine primjene
 - Prije poduzorkovanja niskopropusni filter

- ◆ Poruka se pretvara (*transformira*) u oblik pogodniji za kompresiju
 - npr. iz vremenske u frekvencijsku domenu
- ◆ Transformacija je reverzibilna i ne komprimira
- ◆ Kompresija u drugom koraku: odbacivanje i/ili kvantizacija
 - Transformacija omogućuje kompresiju, premda je izravno ne vrši

Primjer: princip transf. kodiranja (1/3)

TEŽINA (x_0)	VISINA (x_1)
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

$$\mathbf{X} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

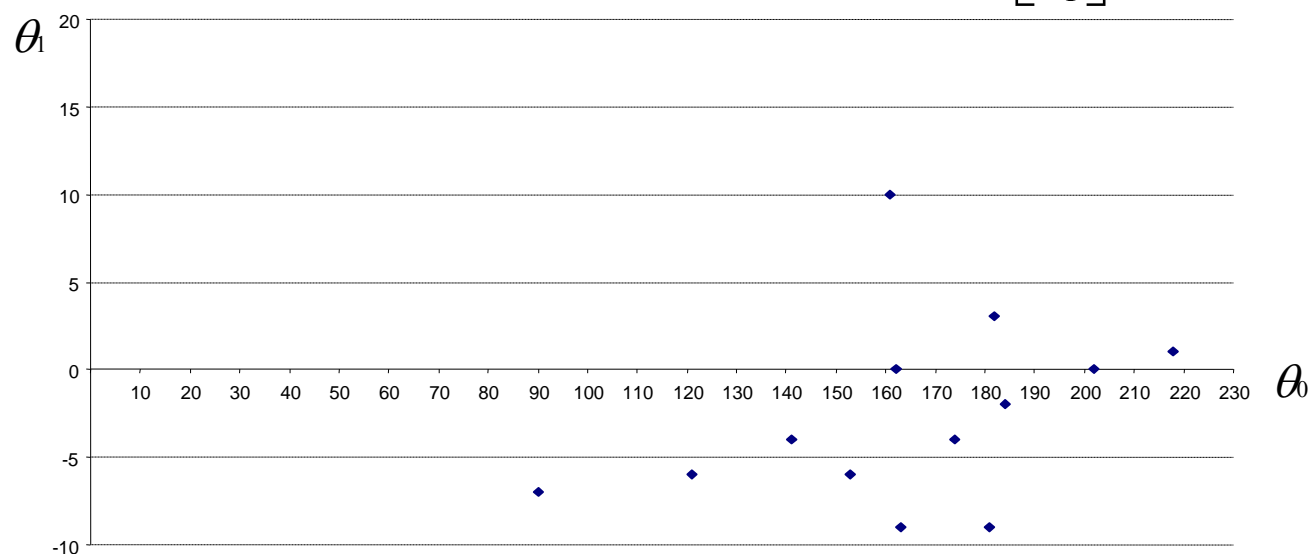


Primjer: princip transf. kodiranja (2/3)

- Rotiramo točke grafa za Φ : $\Theta = AX$

$$\mathbf{A} = \begin{bmatrix} \cos \Phi & \sin \Phi \\ -\sin \Phi & \cos \Phi \end{bmatrix} = \begin{bmatrix} 0.37139068 & 0.92847669 \\ -0.92847669 & 0.37139068 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

θ_0	θ_1
182	3
202	0
162	0
184	-2
141	-4
218	1
174	-4
121	-6
90	-7
161	10
163	-9
153	-6
181	-9
135	-15



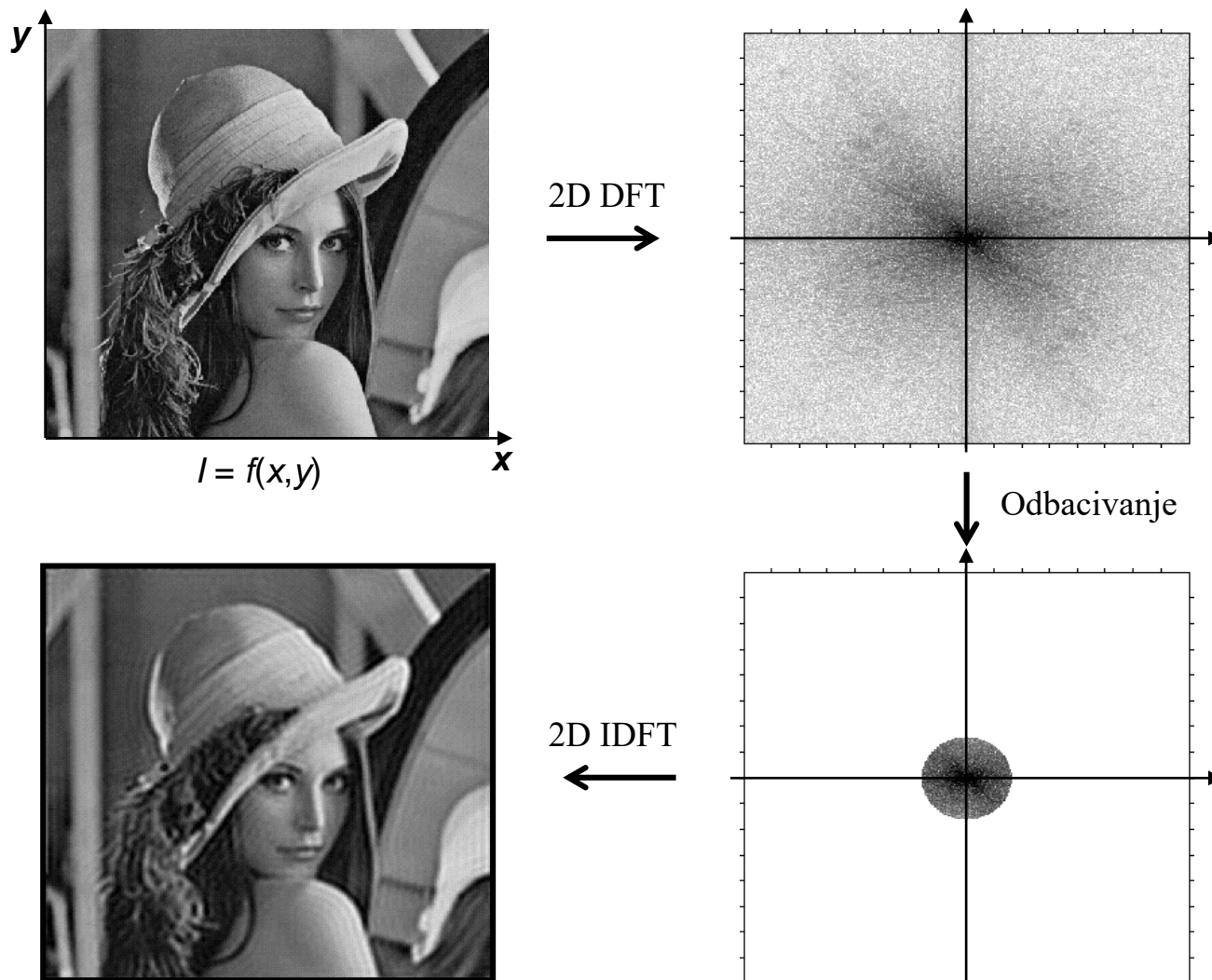
$$\mathbf{A}^{-1} = \begin{bmatrix} \cos \Phi & -\sin \Phi \\ \sin \Phi & \cos \Phi \end{bmatrix} \quad \mathbf{X} = \mathbf{A}^{-1} \Theta$$

Primjer: princip transf. kodiranja (3/3)

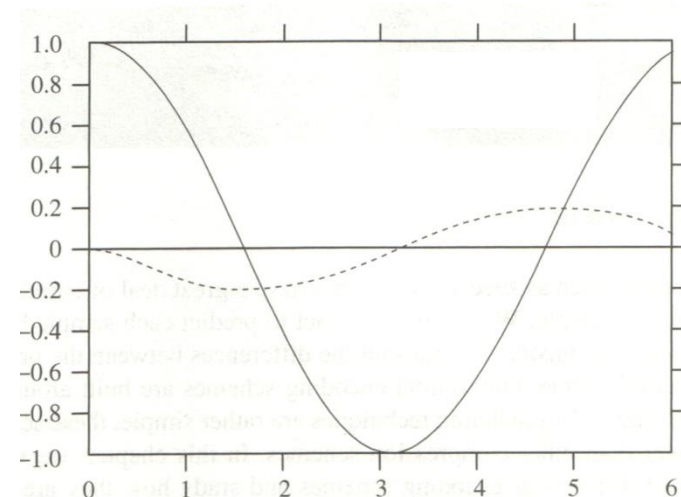
- ♦ Sve θ_1 izjednačimo s nulom (ne treba ih kodirati)
- ♦ Inverzna transformacija: $\mathbf{X} = \mathbf{A}^{-1}\mathbf{\Theta}$

Izvorni podaci		Rekonstruirani podaci	
TEŽINA (X_0)	VISINA(X_1)	TEŽINA (X_0)	VISINA(X_1)
65	170	68	169
75	188	75	188
60	150	60	150
70	170	68	171
56	130	53	131
80	203	81	203
68	160	65	162
50	110	45	112
40	80	34	84
50	153	60	150
69	148	61	151
62	140	57	142
76	164	67	168
64	120	50	125

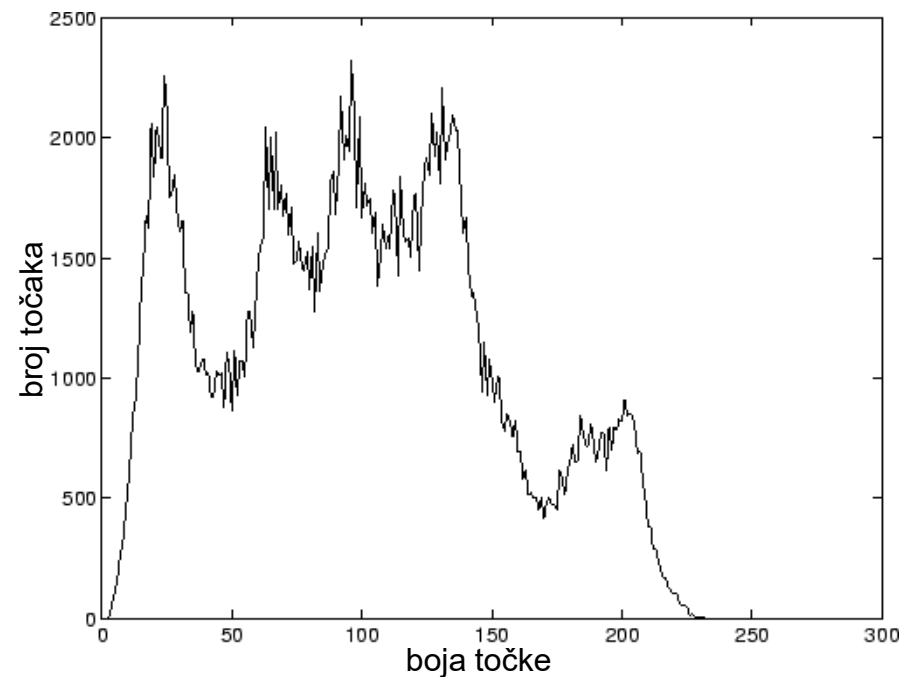
Primjer 2: Fourierova transformacija slike



- ◆ Koristi korelaciju među susjednim uzorcima
 - U vremenu (npr. zvuka) ili prostoru (npr. slika)
- ◆ Svaki uzorak se predviđa iz prethodnih uzoraka
- ◆ Predviđena – stvarna vrijednost = ***signal razlike***
 - Manji raspon i promjena → bolje kodiranje
- ◆ Jednostavno predviđanje: $x_n = x_{n-1}$
 - Signal razlike je razlika među susjednim uzorcima

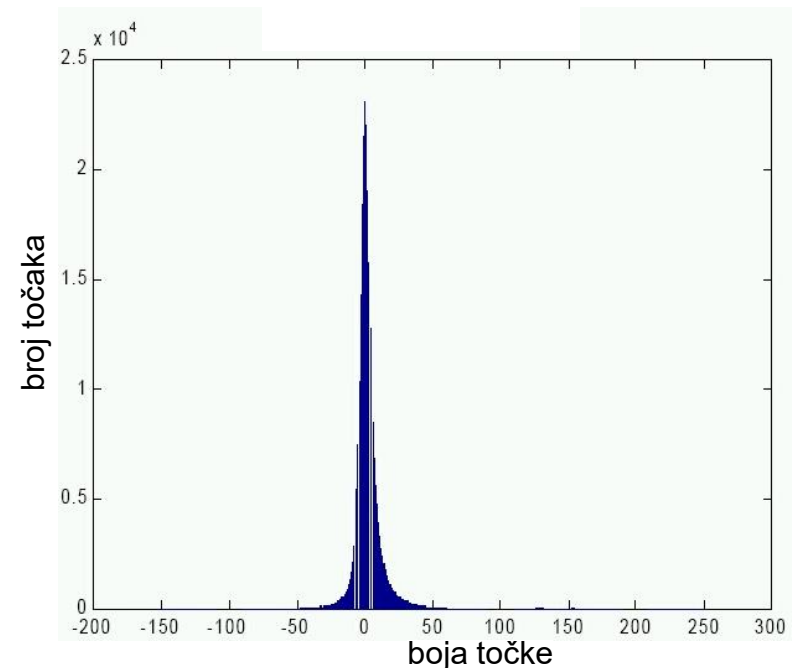
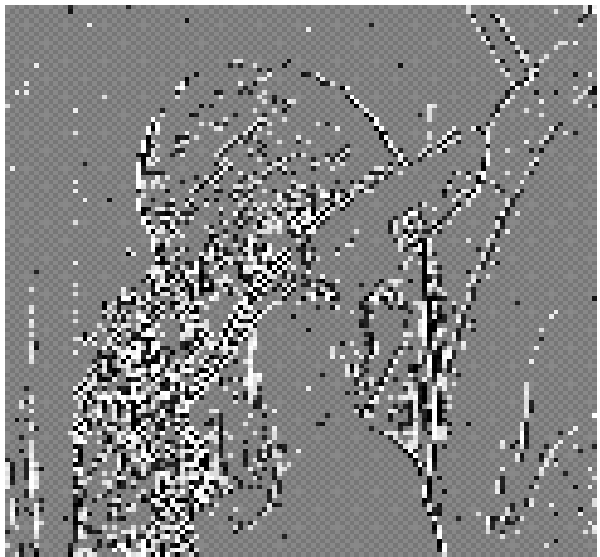


Primjer: diferencijalno kodiranje slike



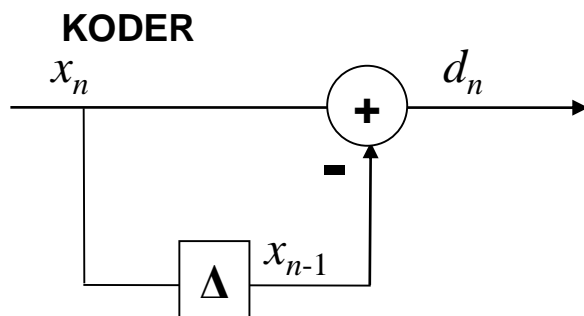
Direktnim entropijskim kodiranjem može se postići 7 bita po točki

- ◆ Razlika susjednih pixela

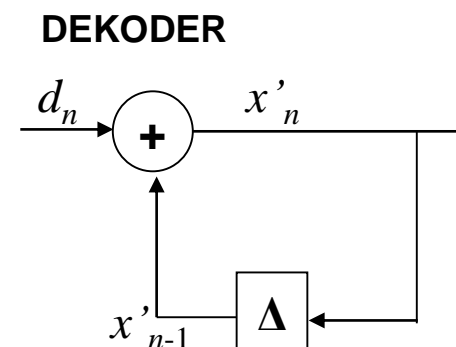


- ◆ Signal se ujednačuje, koncentracija u malom broju vrijednosti, entropija 2.6 bita po točki

Primitivni postupak diferencijalnog kodiranja



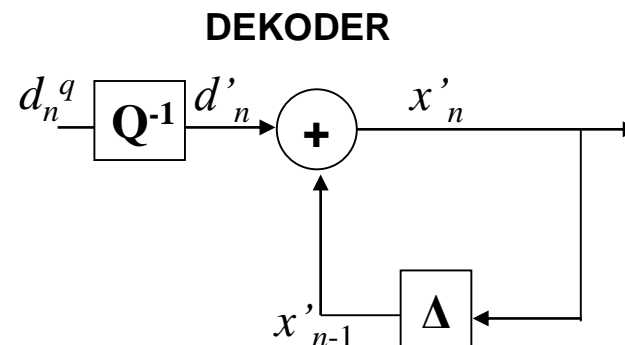
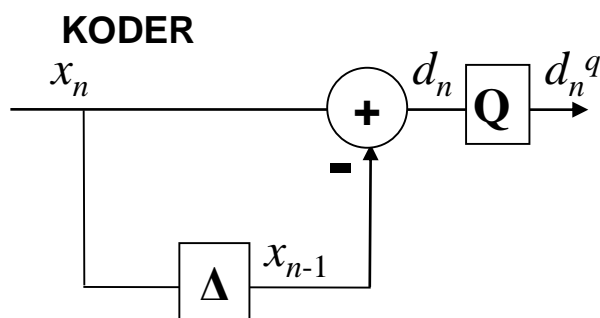
$$d_n = x_n - x_{n-1}$$



$$x'_n = d_n + x'_{n-1} = x_n - x_{n-1} + x'_{n-1} = x_n$$

- ◆ $\{x_n\}$: 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8
- ◆ $\{d_n\}$: 6.2 3.5 3.5 -7.3 2.1 -0.6 -3.2 -2.4
- ◆ $\{x'_n\}$: 6.2 9.7 13.2 5.9 8 7.4 4.2 1.8
- ◆ Pogreška: 0

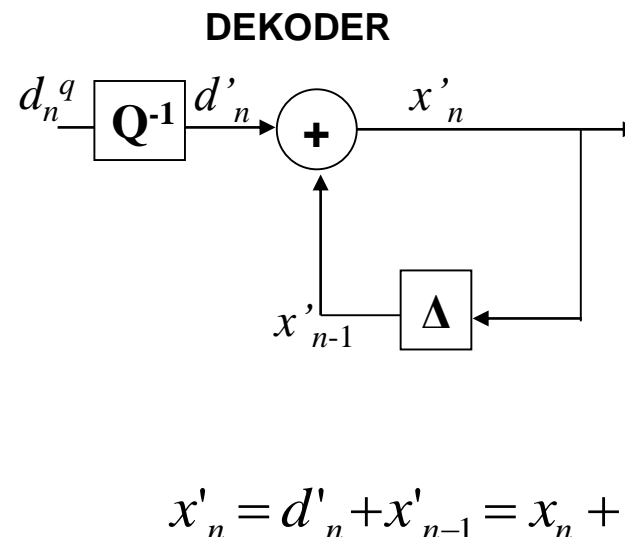
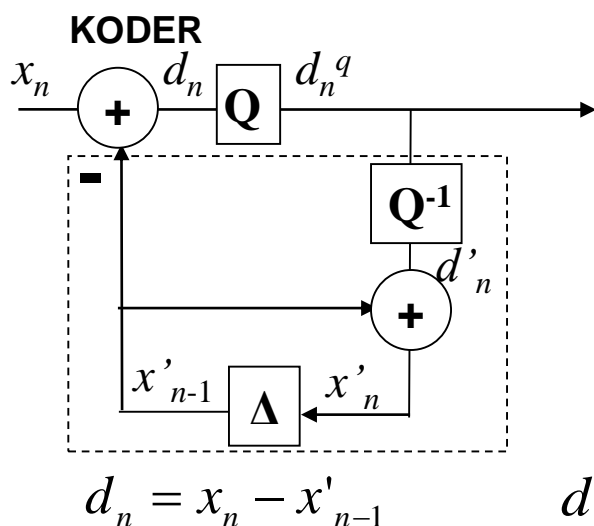
Primitivni postupak sa kvantizacijom



$$d_n = x_n - x_{n-1} \quad d'_n = d_n + q_n \quad x'_n = d'_n + x'_{n-1} = x_n + \sum_{k=1}^n q_k$$

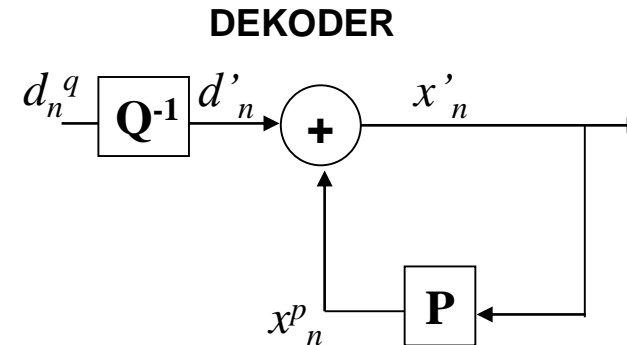
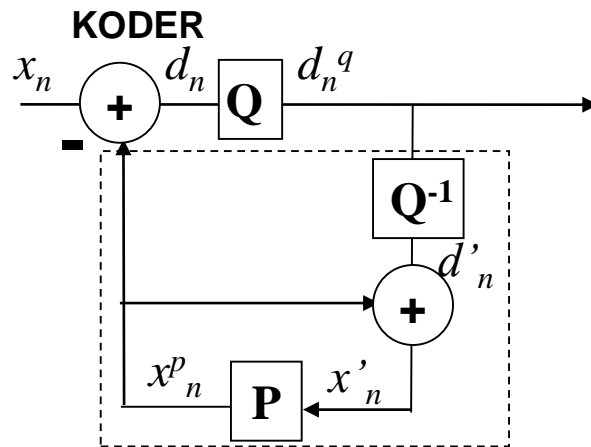
- ◆ Kvantizator sa 7 razina: -6, -4, -2, 0, 2, 4, 6
- ◆ $\{d'_n\}$: 6 4 4 -6 2 0 -4 -2
- ◆ $\{x'_n\}$: 6 10 14 8 10 6 4
- ◆ $\{\varepsilon_n\}$: 0.2 -0.3 -0.8 -2.1 -2 -2.6 -1.8 -2.2
 - Pogreška kvantizacije se akumulira

Izbjegavanje akumulacije kvant. pogreške



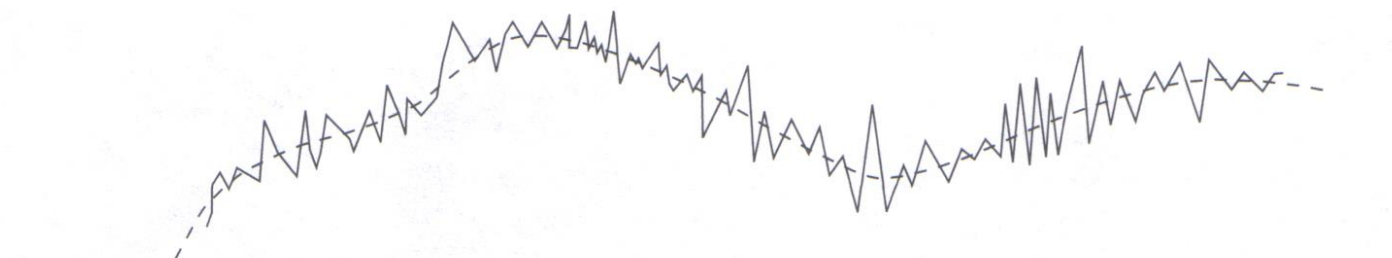
- ◆ $\{d'_n\}$: 6 4 4 -6 0 0 -4 -2
- ◆ $\{x'_n\}$: 6 10 14 8 8 8 4 2
- ◆ $\{\varepsilon_n\}$: 0.2 -0.3 -0.8 -2.1 0 -0.6 0.2 -0.2

Osnovni postupak diferencijalnog kodiranja



- ◆ Prediktor: $x_n^p = P(x_{n-1}, x_{n-2}, \dots, x_{n-k})$
- ◆ Linearna predikcija: $x_n^p = a_1 x_{n-1} + a_2 x_{n-2} + \dots + a_k x_{n-k}$

- ◆ Signal se razdvaja na frekvencijske pojaseve
 - Dobiva se niz signala, po jedan za svaki frek. pojas
 - Svaki ima drugačije karakteristike i važnost
- ◆ Kodiranje svakog pojasa posebno
 - Više ili manje bitova s obzirom na važnost
 - Razne metode kodiranja s obzirom na karakteristike
- ◆ Primjeri primjene: kodiranje zvuka (MP3), slike (JPEG 2000)



Primjer (1/5)

- ◆ Kodiramo niz $\{x_n\}$

10 14 10 12 14 8 14 12 10 8 10 12

- ◆ Diferencijalno kodiranje; signal razlike

10 4 -4 2 2 -6 6 -2 -2 -2 2 2

- ◆ Dinamički raspon -6 do 6 = 12
- ◆ Koristimo m bitova; $M = 2^m$ razina kvantizacije
- ◆ Kvantizacijski interval: $\Delta = 12/M$
- ◆ Maksimalna pogreška kvantizacije: $\Delta/2 = \mathbf{6/M}$

Primjer (2/5)

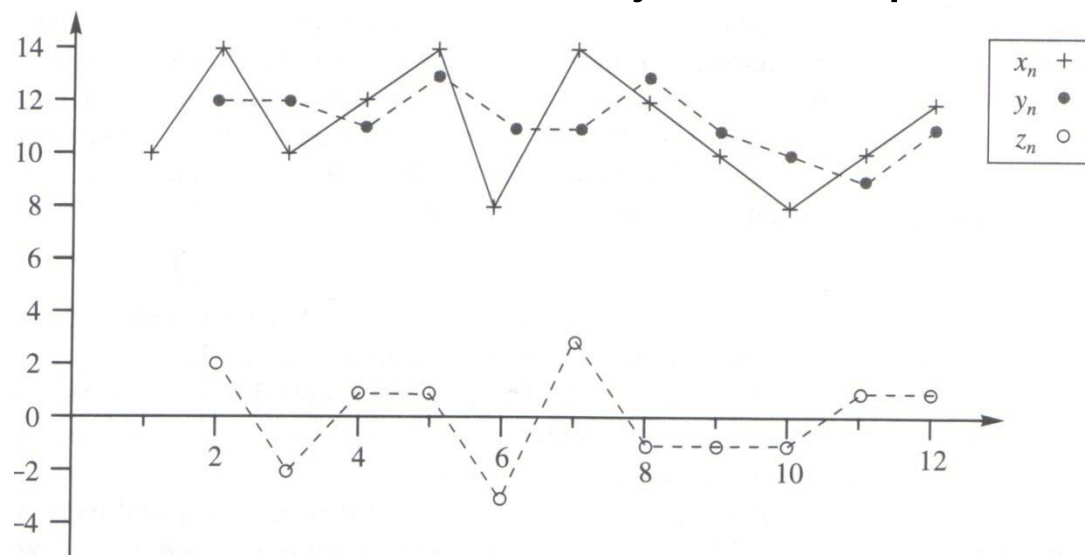
- ◆ Rastavimo niz $\{x_n\}$ na dva nova niza $\{y_n\}$ i $\{z_n\}$

- $\{y_n\}$ je prosjek susjednih vrijednosti – niskofrekvencijska komponenta

$$y_n = \frac{x_n + x_{n-1}}{2}$$

- $\{z_n\}$ je razlika susjednih vrijednosti – visokofrekvencijska komponenta

$$z_n = \frac{x_n - x_{n-1}}{2}$$



$$x_n = y_n + z_n$$

Primjer (3/5)

- ◆ Kodiramo niz $\{y_n\}$

10 12 12 11 13 11 11 13 11 10 9 11

- ◆ Diferencijalno kodiranje; signal razlike

10 2 0 -1 2 -2 0 2 -2 -1 -1 2

- ◆ Dinamički raspon -2 do 2 = 4
- ◆ Koristimo m bitova; $M = 2^m$ razina kvantizacije
- ◆ Kvantizacijski interval: $\Delta = 4/M$
- ◆ Maksimalna pogreška kvantizacije: $\Delta/2 = \mathbf{2/M}$

- ◆ Kodiramo niz $\{z_n\}$

0 2 -2 1 1 -3 3 -1 -1 -1 1 1

- ◆ Signal razlike imao bi veću varijaciju od samog niza, stoga ne koristimo diferencijalno kodiranje nego direktno kvantiziramo
- ◆ Dinamički raspon -3 do 3 = 6
- ◆ Koristimo m bitova; $M = 2^m$ razina kvantizacije
- ◆ Kvantizacijski interval: $\Delta = 6/M$
- ◆ Maksimalna pogreška kvantizacije: $\Delta/2 = \mathbf{3/M}$

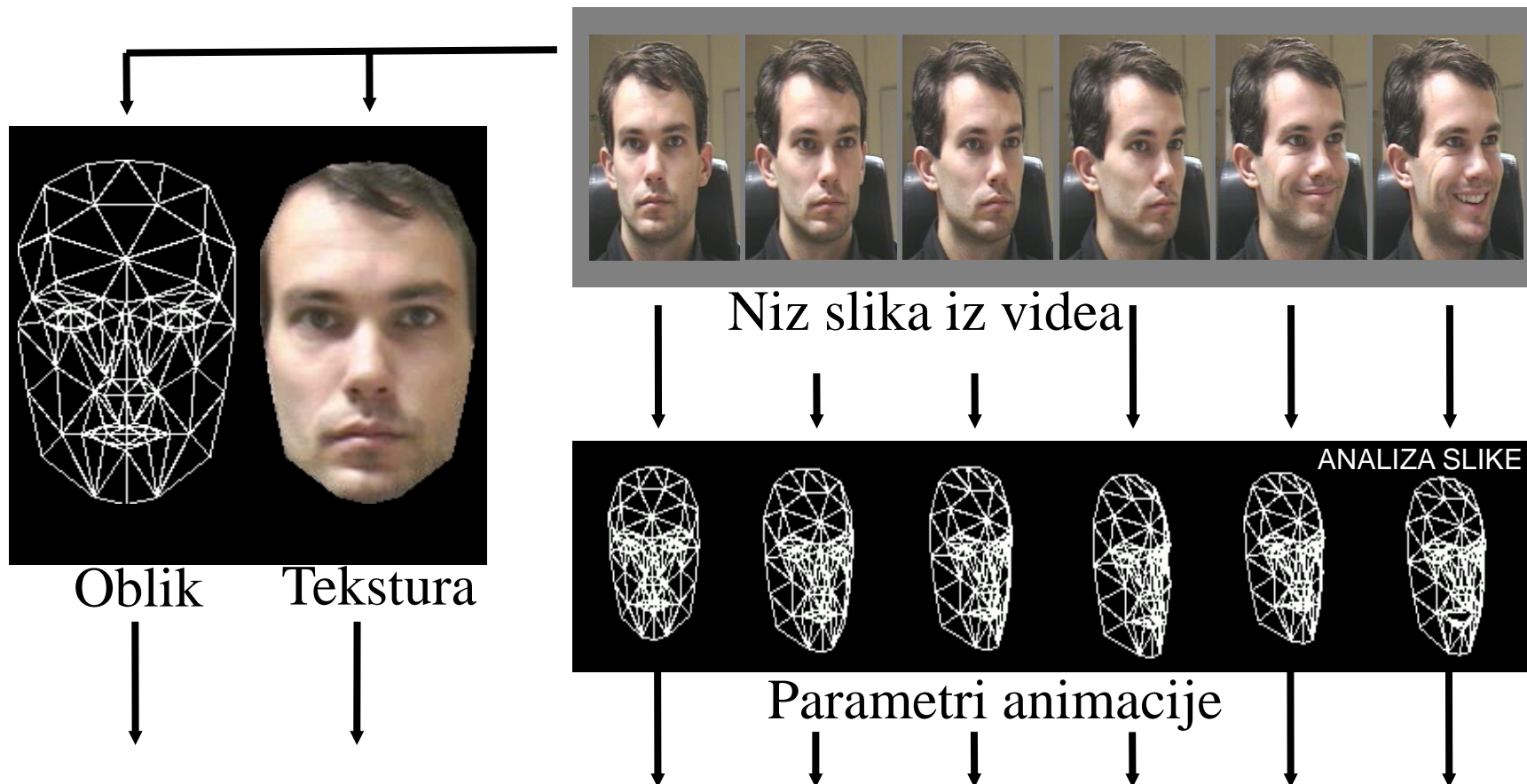
- ◆ Uz jednak broj bitova, max. pogreška kvantizacije
 - Za $\{x_n\} \rightarrow 6/M$
 - Za $\{y_n\} \rightarrow 2/M$
 - Za $\{z_n\} \rightarrow 3/M$

Rastavljanje na komponente = bolje kodiranje!
- ◆ Međutim, kodiramo dva niza umjesto jednog!?
 - Ne! Kodiramo samo svaki drugi član iz $\{y_n\}$ i $\{z_n\}$
 - Iz svakog drugog para y_n i z_n rekonstruiramo dva susjedna člana niza $\{x_n\}$

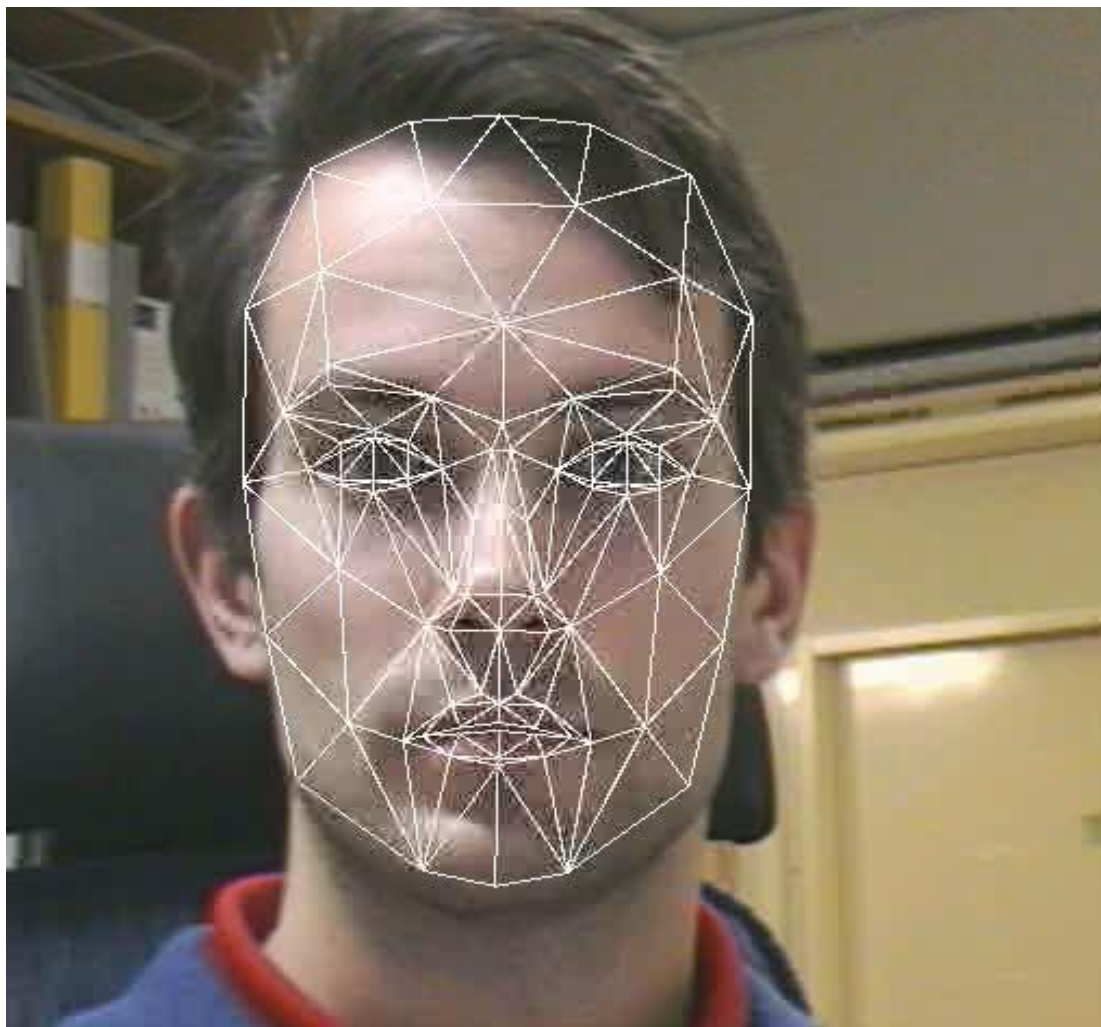
$$x_n = y_n + z_n \qquad x_{n-1} = y_n - z_n$$

- ◆ Ne prenose se uzorci, nego parametri modela
- ◆ Na dekoderu se iz parametara pomoću modela sintetiziraju podaci slični izvornima
- ◆ Primjer
 - Koderi govora zasnovani na modelu (GSM)
 - Fraktalno kodiranje slike
 - Kodiranje videa ljudskog lica

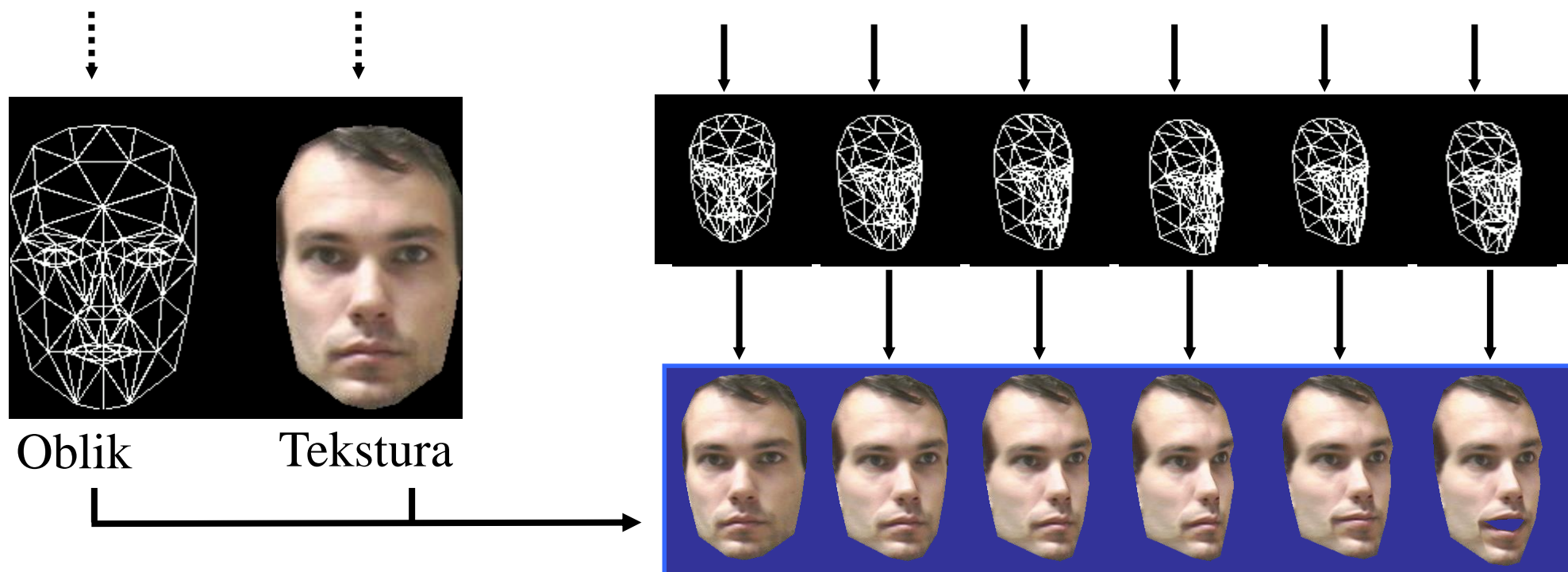
Primjer: video ljudskog lica



Praćenje parametara u videu



Dekodiranje



Rezultat dekodiranja na drugom licu



Još jedan primjer

