

# Video tehnologija

## 14. Zgoščevanje slike informacijska teorija

# Osnove zgoščevanja

- ▶ Slikovne in video informacije zasedejo veliko prostora in pasovne širine za prenos
- ▶ Slika formata A4 v ločljivosti 200 dpi za prenos po faksu zasede 3.7 Mbita
- ▶ Digitalni video ločljivosti 720 x 480 točk, s frekvenco 30 okvirjev/s zasede 20.7 MB/s!
  - na 650MB CD-ROM lahko shranimo v nestisnjeni obliki le 31 s digitalnega video signala

# Osnove zgoščevanja

- ▶ Pri prenosu digitalnega video signala smo omejeni s pasovno širino
  - Ethernet: 10 ali 100 Mb/s
  - ATM omrežje: 100 Mb/s
- ▶ Zahteve po shranjevanju slikovnih informacij
  - Digitalni video
  - Elektronske enciklopedije...

# Primeri avdio in video zgoščevanja

Aplikacija	Pretok podatkov	
	direkten	zgoščen
Telefonski glas	64 kb/s	2-4 kb/s
Video konferenca 352x240, 15 o/s, 8 bitni	30.4 Mb/s	64-768 kb/s
Digitalni avdio 44.1k, 16 bitni	1.5 Mb/s	128k-1.5Mb/s
Digitalni video na CD-ROMU 352x240, 30 o/s, 8 bitni	60.8 Mb/s	1.5-4 Mb/s
HDTV 1280x720, 60 o/s, 8 bitni	1.33 Gb/s	20 Mb/s

## Osnovni pojmi

- ▶ **Zgostitveno razmerje**
  - razmerje med velikostjo originalne in stisnjene slike
- ▶ **bpp** - število bitov na točko (bits per pixel)
  - $\text{bpp} = \text{št. bitov} / \text{št. točk na sliki}$
- ▶ **Podatek**
  - številna vrednost, ki predstavlja svetlost točke
- ▶ **Informacija**
  - smiselna interpretacija podatkov

## Tehnike zgoščevanja

- ▶ **Brezizgubne (brez izgube podatkov)**
  - lahko eksaktno rekonstruiramo originalno sliko
  - zgostitveno razmerje 2:1 do 3:1
  - uporabno za arhiviranje slik (npr. v medicini)
- ▶ **Izgubne (dovoljuje določeno izgubo)**
  - originalne slike ne moremo eksaktno rekonstruirati iz stisnjene slike
  - zgostitvena razmerja 10:1 do 20:1
  - uporabne za TV prenos, video konference, faksiranje dokumentov...

## Kako izvedemo zgoščevanje

- ▶ Osnova je odstranitev redundantnih podatkov
- ▶ Vrste redundance
  - redundanca v kodiranju – kode z variabilno dolžino (RLC, Huffman, aritmetične, LZW)
  - redundanca med točkami – mapiranje (DPCM, DCT, valčna transformacija)
  - redundanca v percepciji – kvantizacija (SQ, VQ)
  - časovna redundanca v video signalu – predikcija gibanja, kodiranje gibanja

## Redundanca v kodiranju

- ▶ Nivoji svetlosti so običajno predstavljeni s kodami enake dolžine (npr. 8 bitne kode)
- ▶ Bolj učinkovito kodiranje dosežemo s kodo variabilne dolžine
- ▶ Kode variabilne dolžine porabijo manj bitov za nivoje, ki se pojavljajo s statistično večjo verjetnostjo

## Primer redundance

$r_k$	$p_k$	koda 1	$l_1$	koda 2	$l_2$
$r_0$	0.19	000	3	11	2
$r_1$	0.25	001	3	01	2
$r_2$	0.21	010	3	10	2
$r_3$	0.16	011	3	001	3
$r_4$	0.08	100	3	0001	4
$r_5$	0.06	101	3	00001	5
$r_6$	0.03	110	3	000001	6
$r_7$	0.02	111	3	000000	6

## Primer redundance

► Povprečna dolžina kode za zapis točke:

$$L = \sum_{k=0}^{L-1} l(r_k) p_k(r_k)$$

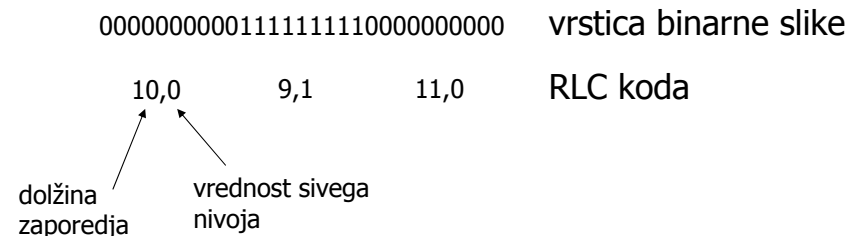
- $l(r_k)$  je dolžina kode za nivo  $r_k$
  - $p_k(r_k)$  je verjetnost nivoja  $r_k$
- Povprečna dolžina kode 1 je 3, povprečna dolžina kode 2 pa je 2.7

## Redundanca med točkami

- Nivoji svetlosti so med sosednjimi točkami v visoki korelaciji
- Nivo svetlosti posamezne točke lahko predvidimo iz nivoja sosednjih točk in za zapis uporabimo le razliko med nivoji
  - preslikava se imenuje mapiranje slike
- Uporabimo lahko kodiranje dolžine (run-length coding)

## Kodiranje dolžine

- Zaporedje točk z enakim nivojem sivine kodiramo z dvojico (št. podatkov, vrednost nivoja)



## Redundanca v percepciji

- ▶ Določene informacije v sliki so za človeško oko bolj pomembne od drugih
- ▶ Oko razločuje frekvence okoli 50 ciklov na stopinjo
  - višje frekvence lahko odstranimo s podvzorčenjem, vendar moramo paziti na kvantizacijske artefakte
- ▶ Barvno informacijo lahko podvzorčimo zaradi manjše občutljivosti očesa na variacije barve

## Kriteriji natančnosti reprodukcije

- ▶ Vernost reprodukcije določamo z dvema vrstama kriterijev:
  - objektivni, temeljijo na matematični meritvi napake na rekonstruirani sliki
  - subjektivni, temeljijo na opazovanju in ocenjevanju slik
- ▶ Objektivni kriterij: RMS napaka

$$e_{RMS} = \sqrt{\frac{1}{MN} \sum_{r=0}^{M-1} \sum_{c=0}^{N-1} (\hat{I}(r,c) - I(r,c))^2}$$

## Objektivni kriteriji

- ▶ RMS razmerje signal šum:

$$SNR_{RMS} = \sqrt{\frac{\sum_{r=0}^{M-1} \sum_{c=0}^{N-1} I(r,c)^2}{\sum_{r=0}^{M-1} \sum_{c=0}^{N-1} (\hat{I}(r,c) - I(r,c))^2}}$$

- ▶ Vršno razmerje signal šum:

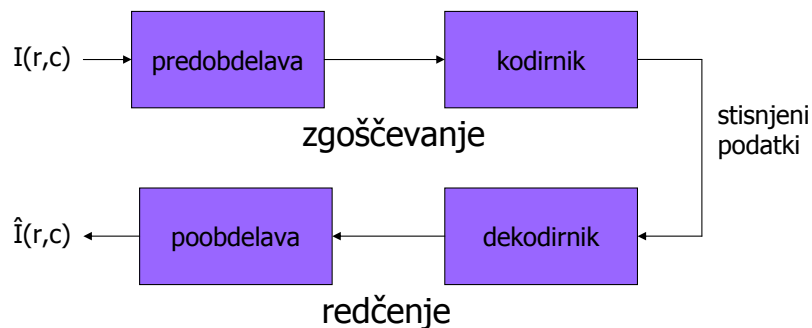
$$SNR_{PEAK} = \sqrt{\frac{(L-1)^2}{\frac{1}{MN} \sum_{r=0}^{M-1} \sum_{c=0}^{N-1} (\hat{I}(r,c) - I(r,c))^2}}$$

## Subjektivni kriteriji

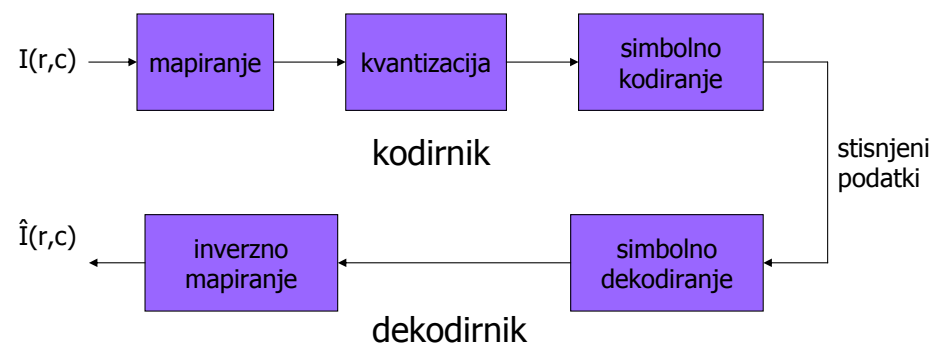
- ▶ Subjektivni kriteriji temeljijo na primerjavi slik
- ▶ Test različnosti
  - ocene od 5-neopazne razlike do 1-preveč moteče
- ▶ Test kvalitete slike
  - ocene od A-odlična do E-slaba kvaliteta
- ▶ Primerjalni testi
  - ocene od +2 precej boljša do -2 precej slabša

## Model zgoščevanja slike

- ▶ Kodirnik odstrani redundance na vhodni sliki



## Kodirnik in dekodirnik



- ▶ Mapirnik odstrani redundanco med točkami
- ▶ Kvantizator odstrani redundanco v precepciji
- ▶ Simbolni kodirnik odstrani redund. v kodiranju

## Informacijska teorija

- ▶ Informacijska teorija je matematična osnova za optimalni zapis podatkov brez izgube informacij
- ▶ Dogodek E, ki se zgodi z verjetnostjo  $P(E)$  vsebuje  $I(E) = \log(1/P(E))$  informacij
- ▶  $I(E)$  je merilo nedoločenosti, ki je v inverzni povezavi z verjetnostjo dogodka

## Informacijska teorija

- ▶ Povprečna informacija v abecedi  $A \in \{a_1, a_2, \dots, a_n\}$  z verjetnostjo simbolov  $P(a_1), P(a_2), \dots$

$$H(z) = -\sum_{i=1}^n P(a_i) \log P(a_i)$$

- ▶  $H(z)$  je nedoločenost ali entropija
- ▶ Entropija določa povprečno količino informacij pri opazovanju enega izhoda
- ▶ Če so simboli enako verjetni, je entropija maksimalna

## Kodiranje z variabilno dolžino

- ▶ Osnovna zamisel je uporabiti najkrajše kode za najbolj verjetne simbole
- ▶ Simboli so lahko nivoji sivine ali pa mapirana slika (točkovne difference, RLC kode...)
- ▶ Uporabljata se dve metodi:
  - Huffmanovo kodiranje
  - aritmetično kodiranje

## Huffmanovo kodiranje

- ▶ Generira optimalno kodo, pri čemer v enem trenutku kodiramo en simbol
- ▶ Koraki kodiranja:
  - izračun porazdelitve verjetnosti simbolov
  - urejanje po verjetnosti
  - reduciranje simbolov: kombiniranje simbolov z min. verjetnostjo v en simbol
  - kodiranje reduciranih simbolov

## Reduciranje simbolov

simbol	verjetnost	1	2	3	4
a <sub>2</sub>	0.4	0.4	0.4	0.4	0.6
a <sub>6</sub>	0.3	0.3	0.3	0.3	0.4
a <sub>1</sub>	0.1	0.1	0.2	0.3	
a <sub>4</sub>	0.1	0.1	0.1		
a <sub>3</sub>	0.06	0.1			
a <sub>5</sub>	0.04				

## Kodiranje simbolov

simbol	verjetnost	koda	1	2	3	4
a <sub>2</sub>	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a <sub>6</sub>	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a <sub>1</sub>	0.1	011	0.1 011	0.2 010	0.3 01	
a <sub>4</sub>	0.1	0100	0.1 0100	0.1 011		
a <sub>3</sub>	0.06	01010	0.1 0101			
a <sub>5</sub>	0.04	01011				

# Video tehnologija

## 15. Morfološke operacije in segmentacija

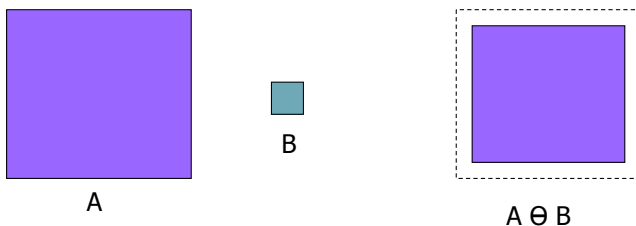
# Morfološke operacije

- ▶ Orodja za ekstrakcijo slikovnih komponent, ki se uporabljajo za opis objektov na sliki
  - robovi objektov
  - skelet
- ▶ Morfološke operacije temeljijo na teoriji matematičnih množic
- ▶ Morfološke operacije se izvajajo običajno na binarnih slikah

# Krčenje in širjenje množice

- ▶ Krčenje in širjenje množice sta dve osnovni morfološki operaciji
- ▶ Krčenje (erozija) množice:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$



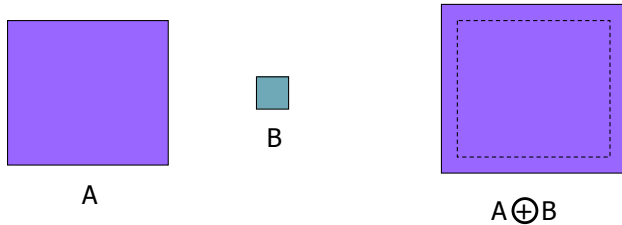
# Krčenje množice

- ▶ Operacijo lahko obravnavamo kot konvolucijo – s premikanjem maske B preko množice A določamo izhodne točke
- ▶ Krčenje uporabljamo za odstranjevanje nepomembnih detajlov na sliki
  - z velikostjo strukturnega elementa (množice B) določimo velikost izbranih detajlov
- ▶ Krčenje deluje podobno kot nizkopasovno sito, le da je izhod binarna slika

## Širjenje množice

- ▶ Širjenje (raztezanje) množice:

$$A \oplus B = \{z \mid ((\hat{B})_z \cap A) \subseteq A\}$$



- ▶ Širjenje uporabljamo npr. za premostitev majhnih razpok v slikovnih objektih

## Odpiranje in zapiranje

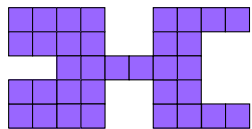
- ▶ Odpiranje je operacija, ki zmehta robove objektov in pobriše ozke prehode in zobce
- ▶ Odpiranje izvedemo s kombinacijo krčenja in širjenja:

$$A \circ B = (A \ominus B) \oplus B$$

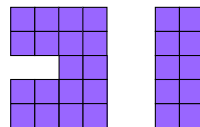
- ▶ Podobno izvedemo zapiranje:

$$A \bullet B = (A \oplus B) \ominus B$$

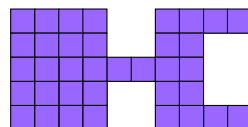
## Odpiranje in zapiranje



Odpiranje:



Zapiranje:



## Morfološki algoritmi

- ▶ S kombinacijo morfoloških operacij naredimo sita za obdelavo binarne slike
- ▶ Osnovna naloga morfoloških algoritmov je ekstrakcija komponent, ki so primerne za predstavitev oblike objektov na sliki
- ▶ Algoritmi:
  - detekcija robov objektov (odštejemo originalno in skrčeno sliko)
  - zapolnjevanje objektov



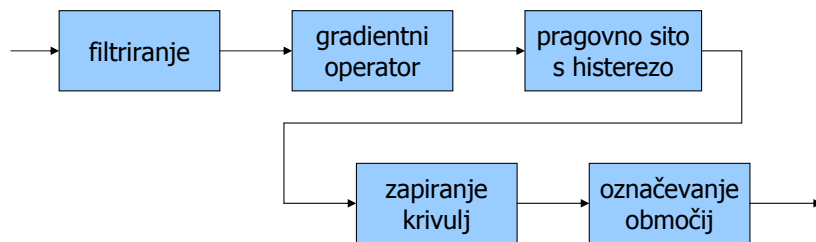
## Segmentacija

- ▶ Segmentacija je postopek, pri katerem sliko razdelimo na sestavne dele in objekte
- ▶ Uporablja se pri avtomatski vizualni kontroli kakovosti izdelkov
- ▶ Segmentacija je v splošnem zelo zahteven postopek, razen v primerih trivialnih slik
  - v industrijskih aplikacijah poskrbimo za dobro razločevanje med objekti in okolico s primerno postavitvijo kamer in osvetlitvijo

## Segmentacija

- ▶ Segmentacija objektov na sliki temelji na eni izmed dveh lastnosti:
  - diskontinuiteti, ki jo predstavljajo robovi objektov
  - podobnosti med točkami objekta na sliki
- ▶ Prva vrsta segmentacije uporablja algoritme za detekcijo robov na sliki
- ▶ Druga vrsta pa temelji na iskanju območij s podobnimi lastnostmi (združevanje regij)

## Segmentacija z detekcijo robov



- ▶ Robovi, ki jih dobimo z gradientom so precej odvisni od izbire praga
  - zaboljšanje rezultata uporabimo pragovno sito s histerezo in algoritem za zapiranje krivulj

## Segmentacija z rastjo območij

- ▶ Slikovne objekte dobimo z združevanjem točk s podobnimi lastnostmi
- ▶ Postopek se začne z množico primarnih točk (semen) okrog katerih združujemo točke
  - kriterij za določitev primarnih točk je odvisen od vrste naloge
- ▶ Postopek končamo, ko zmanjka točk, ki jih lahko vključimo v določeno območje

## Segmentacija z uporabo gibanja

- ▶ Objekte na sliki lahko ločimo od ozadja, če se gibljejo
- ▶ Gibanje detektiramo z uporabo zaporednih slik v video sekvenci
  - primerjamo istoležne točke v različnih časovnih trenutkih
  - gibajoče se dele slike določimo s pragom
  - pri preprosti primerjavi (odštevanju slik) imamo težave s šumom in neenakomerno osvetlitvijo

## Akumulativna diferenca

- ▶ Sliko akumulativne difference dobimo s primerjavo ref. slike z več zaporednimi slikami
  - vsakokrat, ko zaznamo spremembo v istoležnih točkah, povečamo števec sprememb
- ▶ Z akumulativno diferenco učinkovito izločimo naključni šum iz slike