

**UNIVERZA V LJUBLJANI**



**FAKULTETA ZA ELEKTROTEHNIKO**

Jernej Trnkoczy

**IZVORNO KODIRANJE SLIKOVNIH SIGNALOV – IZLOČANJE REDUNDANCE IN  
IRELEVANCE**

**SEMINARSKA NALOGA PRI PREDMETU DIGITALNE KOMUNIKACIJE**

(podiplomski študij)

Ljubljana, November 2006

# I. KAZALO

<b>1</b>	<b>UVOD .....</b>	<b>5</b>
<b>2</b>	<b>ZAJEM IN ZAPIS RASTERSEKE SLIKE .....</b>	<b>7</b>
2.1	Surov zapis rasterske slike .....	7
2.2	Zapisi ki uporabljajo postopke zgoščevanja .....	8
<b>3</b>	<b>REDUNDANCA IN IRELEVANCA V SLIKOVNIH SIGNALIH .....</b>	<b>10</b>
3.1	Redundantna informacija slikovnega signala .....	10
3.1.1	Krajevna redundanca .....	10
3.1.2	Časovna redundanca .....	11
3.1.3	Kodna redundanca .....	12
3.2	Irelevantna informacija slikovnega signala .....	12
3.2.1	Maskiranje kontrasta .....	13
3.2.2	Maskiranje teksture .....	14
3.2.3	Frekvenčno maskiranje .....	14
3.2.4	Časovno maskiranje .....	14
3.2.5	Barvno maskiranje .....	15
<b>4</b>	<b>POSPLOŠEN SISTEM ZA ZGOŠČEVANJE SLIKOVNIH SIGNALOV .....</b>	<b>16</b>
<b>5</b>	<b>IZLOČANJE REDUNDANCE S POMOČJO PRESLIKAV .....</b>	<b>18</b>
5.1	Prediktivno kodiranje .....	18
5.2	Transformacijsko kodiranje .....	20
5.2.1	Diskretna kosinusna transformacija .....	22

5.2.2	Druge pogosto uporabljene transformacije .....	29
<b>6</b>	<b>KVANTIZACIJA .....</b>	<b>31</b>
<b>6.1</b>	<b>Vrste kvantizacije.....</b>	<b>32</b>
6.1.1	Skalarna kvantizacija.....	32
6.1.2	Vektorska kvantizacija.....	35
<b>7</b>	<b>ENTROPIJSKO KODIRANJE SIMBOLOV .....</b>	<b>36</b>
<b>8</b>	<b>NAJPOGOSTEJŠI STANDARDI ZA ZAPIS MIRUJOČE SLIKE.....</b>	<b>41</b>
8.1	Zapis JPEG.....	41
8.2	Zapis JPEG 2000.....	42
8.3	Zapis TIFF .....	43
8.4	Zapis GIF .....	43
8.5	Zapis PNG.....	43
<b>9</b>	<b>ZAKLJUČEK.....</b>	<b>45</b>
<b>10</b>	<b>UPORABLJENA LITERATURA.....</b>	<b>47</b>

## II. SLOVAR UPORABLJENIH KRATIC

DCT	Discrete Cosine Transformation
DFT	Discrete Fourier Transformation
DHT	Discrete Hadamard Transformation
DMS	Discrete Memoryless Source
DST	Discrete Sine Transformation
DWT	Discrete Wavelet Transformation
DWT	Discrete Walsh Transformation
ECQ	Entropy Constrained Quantizer
FFT	Fast Fourier Transform
GIF	Graphics Interchange Format
HDTV	High Definition Television
HSI	Hue Saturation Intensity
HVS	Human Visual System
JPG	Joint Photographic Experts Group
LZW	Lempel-Ziv-Welch
MSE	Minimum Square Error
PCM	Pulse Code Modulation
PNG	Portable Network Graphics
RGB	Red Green and Blue
TIFF	Tagged Image File Format

# 1 UVOD

Izločanje redundance in irelevance iz slikovnega signala je postopek pri katerem skušamo zmanjšati količino podatkov, ki so potrebni za predstavitev slike. Slikovni signal je tako zapisan z manjšim številom bitov kot originalni signal, torej je za prenos takšnega signala potrebna manjša pasovna širina, pri shranjevanju pa potrebujemo manj pomnilniškega prostora. Takemu postopku se običajno reče zgoščevanje oziroma kodiranje (angl. Compression) obratnemu postopku pa dekodiranje ali razgoščevanje (angl. Decompression). Napravi, ki je sposobna vršiti oba postopka pogosto pravimo kodek (angl. Codec - *compression-decompression*).

Zgoščevanje slikovnih signalov je možno, ker le ti vsebujejo določeno mero redundančne informacije. Redundanca se na primer kaže v koreliranosti posameznih slikovnih elementov (angl. Pixel). Verjetnost, da ima skupina slikovnih elementov, ki so si blizu, podobno intenziteto je namreč velika. To pomeni, da so bližnji slikovni elementi močno korelirani, torej iz vrednosti okoliških lahko sklepamo na vrednost danega slikovnega elementa. Druga vrsta redundance nastopa pri predstavitvi posameznih slikovnih simbolov z bitnimi nizi. Pri tem je količina redundančne informacije odvisna od izbranega načina bitne predstavitve signala. Količino izhodnih podatkov lahko zmanjšamo na način, da pogosto pojavljajoče simbole zakodiramo z kratko, redko ponavljajoče pa z dolgo kodno besedo.

Poleg redundance, pri zgoščevanju slikovnega signala izkoriščamo tudi dejstvo, da je potrebna kvaliteta zapisa slikovnega signala odvisna predvsem od njegove uporabe. Vso dodatno informacijo, ki ni nujno potrebna za dobro delovanje neke aplikacije označimo kot irelevantno. Izgubne tehnike zgoščevanja temeljijo prav na odstranjevanju irelevantne informacije. V literaturi se namesto termina irelevance pogosto uporablja termin psiho-vizualna redundanca.

Postopki, ki zmanjšujejo količino podatkov na račun redundance so običajno popolnoma reverzibilni, kar pomeni, da pri procesu zgoščevanja ne pride do izgube informacije. Takšnemu zgoščevanju pravimo brezizgubno zgoščevanje. Postopki, ki zgoščevanje dosežejo na račun irelevance so seveda izgubne narave. Originalne slike iz tako zgoščenih podatkov ne moremo rekonstruirati, predstavimo lahko le njen bolj ali manj dober približek.

Standardi na področju zgoščevanja slikovnih signalov vsebujejo tako brezizgubne kot tudi izgubne postopke. Zavedati se moramo, da z uporabo brezizgubnih postopkov ne moremo doseči željene stopnje zgoščevanja slikovnega gradiva, ki jo zahtevajo sodobne aplikacije. Pri izbiri postopkov zgoščevanje gre tako predvsem za kompromis med kvaliteto ter računsko zahtevnostjo postopka in doseženim faktorjem zgoščevanja, ki ga postopek omogoča.

## 2 Zajem in zapis rasterske slike

### 2.1 Surov zapis rasterske slike

Slika predstavlja stanje svetilnosti okolice, pri čemer tridimenzionalno okolje projiciramo v dvodimenzionalno ravnino – sliko. Slika je tako zvezna dvodimenzionalna funkcija razdalje. Zaradi preslikave, optičnih napak ter šuma, slika nikoli ne predstavlja dejanskega stanja. Z digitalizacijo in zgoščevanjem se izguba informacije še poveča.

Naprave za digitalni zajem slike nam zvezno dvodimenzionalno funkcijo pretvorijo v diskretni dvodimenzionalni prostor. V vsaki diskretni točki prostora, ki predstavlja slikovni element, je podana diskretna vrednost svetilnosti. Naprave za digitalni zajem slik so običajno sestavljene iz dveh enot. Prvo enoto predstavlja senzor občutljiv na določeno področje svetlobnega spektra, pri katerem je izhodna električna veličina proporcionalna energiji (svetilnosti) opazovanega dela spektra. Z takšnimi senzorji »opazujemo« različne točke prostora. Glede na to, ali so opazovane točke med seboj enako oddaljene ali ne, ločimo vzorčenje z enakomerno in vorčenje z spremenljivo razdaljo. Vzorčenje z enakomerno razdaljo je v praksi pogostejše. Dobljene električne signale se nato pretvori (kvantizira) v diskretne vrednosti, te vrednosti pa predstavi z kodnimi besedami. Na ta način dobimo digitalni zapis slike. Kadar v horizontalni smeri opazujemo  $N_1$ , v vertikalni smeri pa  $N_2$  diskretnih točk, kot rezultat dobimo matriko  $N_1 \times N_2$  celih števil, ki predstavlja sliko.

Običajno za surov digitalni zapis monokromatskega slikovnega elementa (enobarvna slika) uporabimo 8 bitov. Na ta način vsak slikovni element opišemo z enim od 256 možnih amplitud dane barve. V primeru črno bele slike imamo na voljo torej 256 sivinskih odtenkov. Za barvni zapis slike najpogosteje uporabljamo zapis s tremi osnovnimi barvnimi komponentami. Znano je, da se da celoten barvni spekter sestaviti iz kombinacijo treh osnovnih barv, na primer rdeče, zelene in modre barve. Na podlagi takšne predstavitve barve deluje zapis RGB (Red Green Blue), pri katerem vsak slikovni element opišemo z tremi vrednostmi, vsaka od njih predstavlja delež posamezne barvne komponente. Seveda lahko barvo posameznega slikovnega elementa zapišemo tudi na drug način, naprimer z komponentami svetlost (angl. Brightness), barvnim odtenkom (angl. Hue) in nasičenostjo (angl. Saturation). Na podlagi takega zapisa se je v Evropi uveljavil model

YUV, v Ameriki pa YIQ. Med posameznimi modeli je možna bijektivna matematična preslikava, kar pomeni, da so si modeli po količini informacije, ki jo nosi slika, enakovredni.

Neglede na izbrani model za predstavitev barve za zapis vsake od treh komponent običajno uporabimo po 8 bitov. To pomeni, da za vsak slikovni element potrebujemo po 24 bitov. Zapis slike, pri katerem je vsak izmed  $N_1 \times N_2$  slikovnih elementov predstavljen z 24 biti, pravimo »surov« (angl. Raw) zapis slike. Tak zapis vsebuje veliko redundantne informacije in je z stališča prenašanja in shranjevanja zelo potraten.

## **2.2 Zapisi ki uporabljajo postopke zgoščevanja**

V prejšnjem poglavju smo spoznali, da je surov rasterski zapis slik z stališča prenašanja in shranjevanja zelo potraten. Tako naprimer za surov zapis barvnega video signala televizijske kvalitete z 30 sličicami dimenzije 640x480 slikovnih elementov na sekundo, potrebujemo več kot 210 Mb pomničniškega prostora za eno samo sekundo trajanja. Vsaka ura tako zapisanega signala torej zavzame okoli 95GB pomnilniškega prostora. Prihajajoča televiziji visoke ločljivosti (HDTV – High Definition Television) z ločljivostjo 1280x720 slikovnih elementov in 60 sličicami na sekundo, v primeru surovega formata zahteva okoli 560GB pomnilniškega prostora na uro trajanja. Na srečo slikovni signali vsebujejo veliko količino redundantne in irelevantne informacije, ki jo lahko odstranimo in tako dosežemo velike faktorje zgoščevanja pri še sprejemljivi kvaliteti signala.

Za prikaz slike zapisane v zgoščenega formatu je potrebno slikovni signal ponovno rekonstruirati. Rekonstruirani signal je lahko povsem identičen prvotnemu signalu, lahko pa predstavlja samo aproksimacijo prvotnega signala. Tako v osnovi ločimo izgubne in brezizgubne algoritme za zgoščevanje. Izgubni algoritmi pridejo v poštev kadar popolna enakost rekonstruirane signala ni zahtevana, torej v primerih, kadar zaznavanje informacije zaradi izgub ni okrnjeno. Vsa dodatna informacija, ki za pravilno interpretiranje ni potrebna, je označena kot irelevantna in se jo zavrže. Za zgoščevanje slikovnih signalov se izgubni algoritmi uporabljajo zelo pogosto, seveda pa obstajajo izjeme. Tako izgubni algoritmi naprimer niso primerni za zgoščevanje medicinskih slik, saj zgoščevanje takih slik na izguben način lahko privede celo do napačne diagnoze. V literaturi lahko zasledimo tudi t. i. zaznavno brezizgubne zgoščevalne postopke. V takih postopkih je zgoščevanje sicer izgubno vendar le do takšne mere, da človeški zaznavni mehanizem ni sposoben razločiti razlike med prvotnim in rekonstruiranim signalom.



Zgoščevalni postopki so lahko statični ali adaptivni. V primeru statičnih algoritmov se postopek zgoščevanja ne spreminja, pri adaptivnem zgoščevanju pa se postopek prilagaja glede na statistične lastnosti vhodnega signala. Dosegljivo razmerje zgoščevanja je namreč odvisno od statističnih lastnosti vhodnega signala, kar še posebej velja za brezizgubno zgoščevanje. Ker se statistika signala od slike do slike razlikuje, je tudi optimalni zgoščevalni postopek od slike do slike različen.

### **3 Redundanca in irelevanca v slikovnih signalih**

Povedali smo že, da zgoščevanje slikovnih signalov temelji na izločanju redundantne in irelevantne informacije, ki jo vsebuje signal. V slikovnih signalih se nahaja več vrst redundantne in irelevantne informacije. V naslednjih poglavjih bom podal različne vrste redundance in irelevance v slikovnih signalih. Na tem mestu je potrebno opozoriti, da se v literaturi irelevanco pogosto označuje tudi kot psiho-vizualna redundanca.

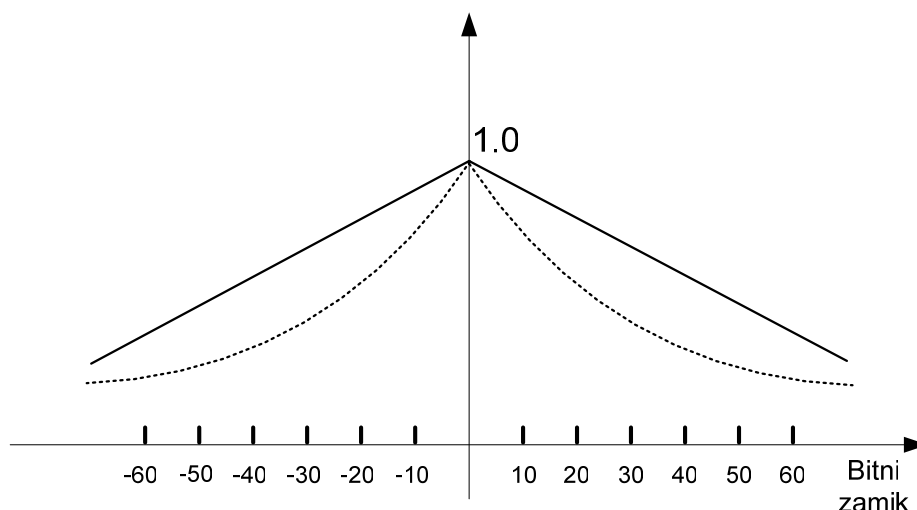
#### **3.1 Redundantna informacija slikovnega signala**

Redundanco, ki se nahaja v slikovnem signalu lahko razdelimo na dve vrsti. Prva vrsta predstavlja redundanco, ki nastopa med slikovnimi elementi slike. Posamezni slikovni elementi mirujoče slike, pa tudi slikovni elementi skupine zaporednih sličic v video signalu, namreč niso statistično neodvisni, ampak so do določene mere korelirani. Ta skupina redundance se naprej razdeli na prostorsko redundanco (Poglavje 3.1.1) in časovno redundanco (Poglavje 3.1.2). Drugo vrsto predstavlja redundanca, ki se nanaša na tehniko dodeljevanja bitnih kodnih besed posameznim slikovnim simbolom in bo opisana v poglavju 3.1.3.

##### **3.1.1 Krajevna redundanca**

Krajevna redundanca (angl. Spatial Redundancy) je posledica statistične korelacije med posameznimi slikovnimi elementi mirujoče slike. Znano je namreč, da so elementi, ki so si krajevno blizu močno korelirani. To v praksi pomeni, da v slikah statistično gledano ni veliko hitrih prehodov v intenziteti slikovnih elementov. Seveda je stopnja korelacije od slike do slike različna. Tako ima slika z veliko detajli manj korelirane slikovne elemente, kot slika z malo detajlov. Velja tudi, da je koreliranost velika pri fotografskih slikah in manjša pri računalniško generiranih grafikah.

Odvisnost med slikovnimi elementi se kaže v avtokorelacijski funkciji posamezne vrstice ali stolpca slikovnih elementov, ki z naraščanjem zamika le počasi upada (Slika 1). Pri slikah z malo detajli je upad skoraj linearen, pri slikah z veliko detajli pa upada hitreje. V obeh primerih velja, da je pri majhnih zamikih avtokorelacija velika.



*Slika 1: Potek avtokorelacijske funkcije za sliko z malo (polna krivulja) in sliko z veliko (črtkana krivulja) detajli.*

Posledica krajevne redundance je, da lahko vrednost intenzitete nekega slikovnega elementa ocenimo na podlagi vrednosti sosednjih slikovnih elementov. To pomeni, da predstavitev vsakega slikovnega elementa v sliki, neodvisno od drugih ni smiselna. Izločanje krajevne redundance iz mirujoče slike je eden od temeljev zgoščevanja slikovnih signalov, dosežemo pa ga lahko na več načinov. To lastnost naprimer izkorišča prediktivno kodiranje (Poglavje 5.1) in kodiranja, ki temeljijo na transformacijah signala (Poglavje 5.2).

### 3.1.2 Časovna redundanca

Časovna redundanca (angl. Temporal Redundancy) nastopa med posameznimi zaporednimi sličicami v video signalu in se torej uporablja pri zgoščevanju videa. V primeru snemanja mirujočih objektov se svetlobne razmere ne spreminjajo, zato se zaporedne sličice ne razlikujejo. V takem primeru nima smisla pošiljati več zaporednih sličic. Prav tako se sličice razlikujejo le malo, če se opazovani objekti gibljejo relativno počasi glede na hitrost zajemanja sličic. Torej so posamezne zaporedne sličice med seboj močno korelirane. Tako lahko naslednjo sličico napovemo iz predhodne/ih, prenašamo pa samo odstopanja od napovedanih vrednosti. Nekatere tehnike predpostavljajo da so spremembe med sličicami v glavnem posledica gibanja objektov in uporabljajo analizo gibanja objektov da povečajo učinkovitost zgoščevanja. Zaradi obsežnosti področja se

bom v tej seminarski nalogo osredotočil predvsem na zgoščevanje mirujočih slik, izločanje časovne redundance pa je dobro opisano v literaturi [1].

### **3.1.3 Kodna redundanca**

Kodna redundanca se razlikuje od redundance, ki smo jo opisali v prejšnjih dveh poglavjih. Ta vrsta redundance se ne nanaša na samo informacijo slikovnega signala, pač pa se ukvarja z predstavitvijo te informacije. Posamezni simboli se v slikovnem signalu namreč pojavljajo z različnimi verjetnostmi. Kadar vse simbole zakodiramo z enakim številom bitov, kot rezultat dobimo večje povprečno število podatkovnih bitov na simbol, kot če pogoste simbole zapišemo z večjim, redko ponavljajoče simbole pa z manjšim številom bitov. Torej zapis, pri katerem so vsi simboli zakodirani z istim številom bitov vsebuje nekaj redundantne informacije. Pri izločanju kodne redundance za nek slikovni signal z podanimi statističnimi lastnostmi izberemo tak zapis posameznih simbolov, ki bo doprinesel čimmanj redundance. Izločanje kodne redundance je predstavljeno v poglavju 7.

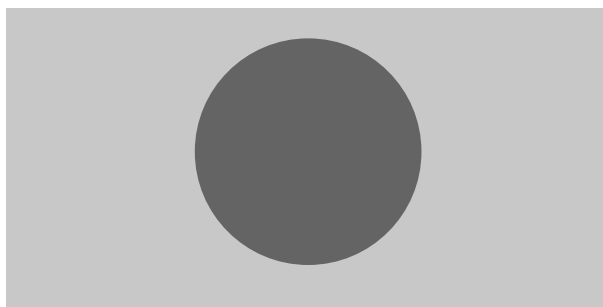
## **3.2 Irelevantna informacija slikovnega signala**

Kot irelevantno lahko označimo vso informacijo, ki za pravilno interpretiranje slikovnega signala ni potrebna. Medtem, ko sta krajevna in časovna redundanca pogojeni z lastnostmi samega signala, pa vzrok irelevantnosti slikovne informacije tiči v karakteristiki človeškega vida (HVS – Human Visual System). Znano je, da je vizualno zaznavanje okolice pri človeku zelo kompleksen mehanizem. Glede na to, da zajeto informacijo obdelujejo človeški možgani to ni presenetljivo, saj njihovo delovanje še danes ostaja v večji meri nepojasnjeno. Vemo, da odziv sistema na vizualno stimulacijo ni linearno odvisen od moči fizikalnih parametrov stimulacije, kot sta to na primer intenziteta ali barva. Človeško zaznavanje je v tem pogledu drugačno od zajemanja slike z senzorjem fotoaparata ali kamere. Nekateri parametri stimulacije očesa so bolj pomembni kot drugi, kar pomeni, da je del informacije, ki jo nosi slika, lahko bolj pomemben kot preostala informacija. Poleg tega med parametri stimulacije očesa, ki so si blizu v času ali prostoru, prihaja do uničujoče interakcije. Taka interakcija lahko privede do napak pri prepoznavanju. Pri preučevanju interakcije opazujemo zmožnost detekcije ene fizikalne stimulacije pri hkratni prisotnosti neke druge stimulacije.

Če iz slikovnega signala odstranimo del informacije, ki je za človeško interpretiranje slikovnega signala nepomembna, potem to ne bo vplivalo na kvaliteto delovanja aplikacije, ki ji je dani slikovni signal namenjen. V takem primeru lahko rečemo, da je del informacije, ki jo nosi prvotni signal irelevanten oziroma, kot navajajo nekateri avtorji, psiho-vizualno redundanten.

### 3.2.1 Maskiranje kontrasta

Maskiranje kontrasta (angl. Luminance Masking) je najizrazitejši izmed vseh poznanih psiho-vizualnih efektov. Dobro znan je primer kroga v nekem sivinskem odtenku na podlagi obarvani z drugim sivinskim odtenkom (Slika 2).



*Slika 2: Vpliv svetlosti podlage na zaznavanje objekta.*

V tem primeru preučujemo vpliv svetlosti ozadja na zmožnost prepoznavanja kroga. Prvo stimulacijo nam torej predstavlja sivina ozadja drugo pa sivina kroga. V primeru strojne obdelave seveda odtenek sivine ozadja nima vpliva na zmožnost detekcije. Pri dovolj veliki razliki med odtenkoma bo sistem zaznal krožnico neglede na odtenek ozadja. V primeru človeškem zaznavanju pa ni tako. Dokazano je, da pri isti razliki med sivinskima odtenkoma krog od ozadja človek loči veliko lažje, kadar sta ozadje in krožnica temna. Povedano z drugimi besedami: kadar je ozadje svetlo za zaznavo kroga človek potrebuje večjo razliko med sivinskima odtenkoma, kot pri temnem ozadju. To je tudi vzrok, da je šum v sliki veliko bolj opazen na temnih, kot pa na svetlih delih slike. V poglavju o kvantizaciji (Poglavje 6) bom pokazal, da kvantizacija signala vnaša kvantizacijski šum.

Nekateri načini kvantizacije upoštevajo zgoraj opisani vizualni efekt za zmanjševanje popačenja slike, ki nastane kot posledica kvantizacije.

### **3.2.2 Maskiranje teksture**

Maskiranje teksture (angl. Texture Masking), ki mu pogosto pravimo tudi krajevno maskiranje (angl. Spatial Masking) je prav tako značilnost človeškega vizualnega zaznavnega sistema. Človeško oko je namreč sposobnejše zaznati male razlike v svetlosti na področjih, ki so približno enakomerne svetlosti, kot pa na področjih, kjer je veliko detajlov. Kot posledica človeško oko šum v sliki lažje opazi na enakomerno svetlem področju, kot pa na delu slike kjer je veliko detajlov. V tem primeru gre torej vpliv teksture slike na sposobnost zaznavanja sivinskih odtenkov. Tudi to lastnost se izkorišča v nekaterih vrstah kvantizatorjev, v težnji po čimmanjši kvantizacijski napaki.

### **3.2.3 Frekvenčno maskiranje**

Pri frekvenčnem maskiranju (angl. Frequency Masking) gre za to, da človeško oko deluje kot nizkopasovni filter. To pomeni, da je bolj občutljivo na nizke, kot na visoke prostorske frekvence v sliki. Povedano drugače: pri hitrem krajevnem spreminjanju sivinskih odtenkov, bo morala biti razlika med njimi večja, kot pri počasnem spreminjanju z naraščajočo dimenzijo, da bomo razlike opazili. V tem primeru gre za vpliv prostorske frekvence slike na zaznavanje razlik sivinskih odtenkov. Ta pojav se naprimer izkorišča po transformaciji z diskretnim kosinusnim transformom (Poglavje 5.2.1). Po transformaciji se zavrže nekatere koeficiente, ki predstavljajo visoke frekvence in na ta način dosežemo zgoščevanje, ne da bi se pri tem kvaliteta slike opazno zmanjšala.

### **3.2.4 Časovno maskiranje**

Časovno maskiranje (angl. Temporal Masking) se navezuje na spreminjanje slike z časom. Pri hitrih časovnih spremembah slike človeško oko potrebuje nekaj časa, da se privadi na novo sliko. Med tem časom človeško zaznavanje ni občutljivo na detajle v sliki. V tem primeru gre torej za vpliv časa med sličicama videa na zaznavanje razlik sivinskih odtenkov.

### 3.2.5 Barvno maskiranje

Barva kot zaznavanje vidne svetlobe je določena z valovno dolžino elektromagnetnega valovanja. Seveda ima vsaka barva tudi intenziteto oz. moč. Vidna svetloba je torej podana z parametrom intenzitete in naborom različnih valovnih dolžin, iz katerih je sestavljena (čemur pravimo krominanca). V krominanci se nahajata dva parametra: barvni odtenek (angl. Hue) in nasičenje (angl. Saturation). Barvni odtenek nam določa dominantna valovna dolžina v naboru valovnih dolžin svetlove, medtem ko je nasičenje določeno z tem kakšen je procent ostalih valovnih dolžin v naboru. Nasičenje nam torej določa kako čiste barve je neka svetloba. Ena sama valovna dolžina ima 100% nasičenje, medtem ko ima bela svetloba nasičenje enako 0%.

Za zapis barvnih sliko poznamo različne načine. Najbolj poznan je zagotovo model RGB (Red Green Blue), saj je izhod naprav za zajemanje slik običajno zapisan z tem modelom. V napravi za zajemanje so namreč običajno tri vrste senzorjev, ki so občutljivi na plavo, rdečo in zeleno komponento spektra. Sliko lahko zapišemo tudi z modelom HSI (Hue Saturation Intensity). Pri pretvorbi nastopa problem praktične narave – iz modela RGB je težko izračunamo vrednosti modela HSI. Zaradi tega so se razvili tudi drugi modeli, kjer vsako od treh komponent izračunamo po nekem enostavnem matematičnem postopku (modeli YUV, YIQ, YDrDb, YCrCb). V vseh navedenih modelih Y komponenta predstavlja svetilnost. Taka predstavitev je pomembna, če želimo izkoristiti lastnost barvnega maskiranja. Človeški zaznavni sistem je namreč bolj občutljiv na spremembe intenzitete kot pa na spremembe barve. Torej je komponenta svetilnosti (ang. Luminance) bolj pomembna kot ostali dve komponenti. Tako je komponento Y smiselno zakodirati z večjim številom bitov, kot ostali dve komponenti. Nekateri sistemi tako komponento Y vzorčijo z dvakrat večjo ločljivostjo, kot ostali dve krominančni komponenti.

## 4 Posplošen sistem za zgoščevanje slikovnih signalov

V procesu načrtovanja sistema za zgoščevanje slikovnega gradiva se trudimo, da bi kar najboljše izločili v prejšnem poglavju opisano redundanco in irelevanco, pri tem pa dosegli kar najmanjšo napako pri rekonstrukciji iz zgoščenega zapisa. V splošnem sistem za zgoščevanje sestoji iz treh glavnih stopenj (Slika 3).



*Slika 3: Blokovni diagram posplošenega zgoščevalnega sistema.*

Prva stopnja vrši izločanje redundance iz vhodnega signala. Odstranitev oz. zmanjšanje redundance se običajno doseže z preslikavo signala iz originalne oblike predstavitve (matrika slikovnih elementov z podanimi intenzitetami) v neko drugo predstavitev. Pogosto uporabljane preslikave so preslikave z pomočjo predikcije posameznih vzorcev na podlagi različnih modelov (Poglavje 5.1), preslikave signala iz krajevnega v frekvenčni prostor, kot je na primer diskretna kosinusna transformacija (Poglavje 5.2.1), dekompozicija originalnega signala na posamezne podpasove, kot na primer diskretna valjčna transformacija itd. Z tem korakom dosežemo kompaktnejšo predstavitev informacije, ki jo nosi originalni signal in sicer v smislu zmanjšanja števila koeficientov, z katerimi je signal predstavljen. Take transformacije zmanjšajo entropijo originalnega signala. Ta korak je večinoma popolnoma reverzibilen, kar pomeni, da v postopku ne pride do izgube informacije.

Naslednji korak v procesu zgoščevanja je nadaljne zmanjševanje entropije preslikanega signala, ki ga doseže z zavrženjem za dano aplikacijo nepomembne informacije prisotne v sliki. Ta korak ni reverzibilen, saj zavržene informacije ne moremo ponovno rekonstruirati, torej se ta korak uporablja le v sistemih, ki dopuščajo izgubno zgoščevanje. Zavrženje



nepomembne informacije se ponavadi doseže z pomočjo kvantizacije (Poglavje 6), pa tudi z zavrženjem nepomembnih koeficientov transformacije po preslikavi signala.

Tretji korak obsega entropijsko kodiranje kvantiziranih koeficientov transformiranega signala. Z dobrim entropijskim kodiranjem se lahko približamo teoretični meji, ki zahteva toliko informacijskih bitov za zapis enega simbola, kot je entropija signala ki ga zgoščujemo (Poglavje 7). Tudi ta korak je popolnoma reverzibilen in ne prinaša izgube informacije.

Redčenje (angl. Decompression) predstavlja obraten proces zgoščevanju. Zgoščen bitni niz je najprej entropijsko dekodiran, kot rezultat dobimo kvantizirane koeficiente signala. Nato sledi inverzna kvantizacija, katere rezultat so aproksimirane vrednosti transformiranih koeficientov. Nato sledi še inverzna transformacija.

## 5 Izločanje redundance s pomočjo preslikav

Z pomočjo preslikav slikovnega signala izločamo krajevno redundanco (Poglavje 3.1.1). Pri tem postoku torej izkoriščamo dejstvo, da so slikovni elementi med seboj korelirani. Vrednost nekega slikovnega elementa v dani sliki statistično namreč ni neodvisna od vrednosti okoliških slikovnih elementov. Obstaja veliko različnih postopkov preslikave signala. V tem poglavju bomo opisali preslikave, ki se pri zgoščevanju slikovnega signala uporabljajo najpogosteje.

### 5.1 Prediktivno kodiranje

Zaradi statistične odvisnosti med slikovnimi elementi slikovni signal torej ni diskretni izvor brez spomina ampak t.i. Markov izvor (angl. Markov source). Posamezne intenzitete slikovnih elementov imajo približno enake verjetnosti nastopanja. Po enačbi za izračun

entropije ( $H = -\sum_{i=1}^n p(s_k) \log_2 \left( \frac{1}{p(s_k)} \right)$ )

(Enačba 8)) je entropija takega signala visoka. V primeru, da tak signal direktno entropijsko kodiramo, ne moremo doseči visokega faktorja zgoščevanja. Lahko pa izkoristimo dejstvo, da se nek slikovni element zelo verjetno ne bo veliko razlikoval od okoliških. Po nekem izbranem modelu torej lahko napovemo vrednost naslednjega slikovnega elementa, izmerimo odstopanje od napovedane vrednosti, ki nam predstavlja napako modela in namesto vrednosti slikovnega elementa zapišemo napako. Dekodirnik po enakem modelu izračuna (napove) vrednost in od nje odšteje zapisano napako. Na ta način dobimo rekonstruirani signal.

Z takšnim procesom seveda nismo dosegli nikakršnega zgoščevanja, za zapis napake namreč potrebujemo celo večje število bitov. Če intenziteta slikovnih elementov lahko zavzema vrednosti od 0 do 255 bo model napovedal neko vrednost iz tega intervala. Možne vrednosti napake pa bodo v razponu od -255 do +255 (napovemo 255 ko je prava vrednost 0 ali napovemo 0 ko je prava vrednost 255). Torej za zapis napake v tem primeru potrebujemo 9 bitov, kar je več kot potrebno število bitov za zapis originalnega signala. Trik je v tem, da bo ob solidnem modelu za napovedovanje v veliki večini primerov napaka majhan. Torej smo iz originalnega signala, ki je imel precej enakomerno verjetnostno

funkcijo (vse vrednosti nastopajo z približno enako verjetnostjo) dobili signal, pri katerem z veliko verjetnostjo nastopajo majhne vrednosti, to pa pomeni signal z majhno entropijo.

Vemo torej, da bo napaka z veliko verjetnostjo majhna, ni pa razloga, da bi bila napaka pri nekem slikovnem elementu po velikosti podobna napakam pri sosednjih slikovnih elementih. Z transformacijo originalnega signala, ki je imel močno korelirane slikovne elemente, smo dobili signal, ki ima precej neodvisne koeficiente (vrednosti napake) in majhno entropijo. Če bi bila transformacija idealna bi začetni Markov izvor, ki ima visoko entropijo preslikali v diskretni signal brez spomina z nizko entropijo. Signal z majhno entropijo je možno zelo učinkovito zakodirati z entropijskim kodirnikom. Torej mora elementu, ki vrši predikcijo slediti entropijski kodirnik. V večini sistemov se med njima pojavlja še kvantizator, ki kvantizira koeficiente napake. Za doseganje zgoščevanja mora sistem vsebovati vse naštet elemente, predikcija sama po sebi le poveča količino podatkov.

Modelov, ki na podlagi sosednjih napovejo vrednost danega slikovnega elementa obstaja veliko. Primer pogosto uporabljenega prediktorja tretjega reda nam v povezavi z sliko Slika 4 predstavlja enačba  $x = 0.75A - 0.5B + 0.75C$  (Enačba 1).

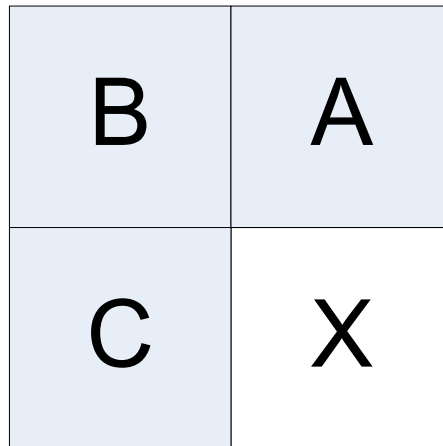
$$x = 0.75A - 0.5B + 0.75C$$

(Enačba 1)

Glede na to, koliko sosednjih elementov upoštevamo pri predikciji, ločimo prediktorje prvega, drugega, tretjega itn. reda. Experimentalno je bilo pokazano, da minimalna srednja kvadratična napaka (Mean Square Error - MSE) močno pade kadar namesto enega vzamemo tri sosednje slikovne elemente, nakar se padanje umiri in se povečevanje upoštevavanja okoliških elementov zaradi računske kompleksnosti na izplača več.

Poseben primer sistema z predikcijo je dobro poznana delta modulacija. To je v bistvu prediktor prvega reda, pri katerem je vrednost konstante kar enaka 1. Torej je napovedani slikovni element kar enak predhodnemu, napako pa opišemo le z dvema možnima

vrednostima ( $+\Delta/2$  ali  $-\Delta/2$ ). Kadar je slikovni element svetlješi kot njegov sosed, je vrednost napake enaka  $+\Delta/2$ , če pa je temnejši je napaka enaka  $-\Delta/2$ .



Slika 4: Prediktor tretjega reda izračuna vrednost  $X$  iz treh sosednjih slikovnih elementov  $A, B$  in  $C$ .

V nekaterih sistemih za zgoščevanje se uporabljajo tudi adaptivni prediktorji. V teh sistemih je predhodno definiran cel nabor različnih prediktorjev (enačb po katerih se bo izračunala napovedana vrednost), nato pa sistem preklaplja med njimi glede na karakteristike vhodnega slikovnega signala.

Kot smo videli sistemi z predikcijo delujejo neposredno v krajevnem prostoru in signala ne preslikujejo v frekvančni prostor. Predikcija je najpogosteje uporabljen postopek za de Korelacijo slikovnih elementov v krajevnem prostoru. Obstajajo pa tudi drugi postopki za de Korelacijo slikovnih elementov v krajevnem prostoru. Od bolj zanimivih omenimo take, ki izkoriščajo samopodobnosti v sliki in jim pravimo fraktalni kodirniki.

## 5.2 Transformacijsko kodiranje

V prejšnjih poglavjih smo opisali posplošen kodirnik slikovnega signala, ki je ponavadi sestavljen iz treh enot: transformacijske, kvantizacijske in enote za entropijsko kodiranje. Transformacijska enota na izhodu daje neko alternativno predstavitev signala, ta pa se nato kvantizira in entropijsko zakodira. V primeru prediktivnega kodiranja (Poglavje 5.1) transformacijska enota deluje direktno v krajevnem prostoru. Ta enota nam na račun

zmanjšanja korelacije med slikovnimi elementi generira signal z majhno entropijo. Pri transformacijskem kodiranju gre za podobno stvar. Kadar so transformirane vrednosti signala manj korelirane, kot vrednosti slikovnih elementov pred transformacijo, bomo z kvantizacijo in entropijskim kodiranjem dosegli velike faktorje zgoščevanja.

Matematično gledano je transformacija pravilo, ki nabor vrednosti preslika v nek drug nabor vrednosti. Preslikava iz prostora RGB v prostor YUV, ki smo jo omenili v poglavju 3.2.5, nam torej že predstavlja neke vrste transformacijo. Rezultat transformiranja je na drug način predstavljena informacija. Matriko vrednosti slikovnih elementov nam preslika v nek drug nabor vrednosti, ki jim običajno rečemo koeficienti transforma. Kadar imajo te koeficienti s stališča zgoščevanja ugodne lastnosti, je transformacija dobra. Ugodne lastnosti koeficientov pomenijo:

- Koeficienti, ki jih dobimo po transformaciji so med seboj čimbolj nekorelirani;
- Transformacija prerazporedi energijo signala v domeni transforma tako, da je večina energije zbrana v relativno majhnem številu koeficientov. Ostali koeficienti bodo tako po velikosti zelo majhni in jih bomo lahko zavrgli, ne da bi pri tem povzročili veliko popačenje slike.

Transformacija je v splošnem lahko izgubna ali neizgubna. Za zgoščevanje slik se običajno uporabljajo breizgubne transformacije. Z samim transformiranjem slikovnega signala torej še ne dosežemo zgoščevanja. Transformiranje tako pomeni samo preoblikovanje signala v obliko, ki jo lahko bolj učinkovito kvantiziramo in entropijsko kodiramo.

Postopek transformacijskega kodiranja navadno poteka v več fazah. V prvi fazi sliko razdelimo v posamezne bloke (podslike). Vsak tak blok nato linearno transformiramo, tako dobimo nabor transformacijskih koeficientov. Tiste koeficiente, ki bistveno ne prispevajo pri rekonstrukciji signala ponavadi zavrzemo. S tem sicer zavrzemo del informacije, vendar le tisto, ki ni bistvenega pomena za rekonstrukcijo signala. Koeficiente, ki jih ne zavrzemo se nato običajno kvantizira in na koncu še entropijsko zakodira.

Velikost blokov (podslik), v katere originalno sliko pred transformacijo razdelimo je pri različnih algoritmi za zgoščevanje različna. Čimvečja je podslika, temvečjo dekorelacijo slikovnih elementov lahko z transformacijo dosežemo. Velja pa tudi, dosežena

dekorelacija z večanjem podslike na začetku raste zelo hitro, nato pa vedno počasneje. Za veliko večino slik velja, da slikovni elementi, ki sta krajevno narazen več kot 20 slikovnih elementov praktično nista več korelirana. Poleg tega transformacija velike podslike pomeni tudi veliko računsko kompleksnost, pojavljajo pa se tudi drugi vrste problemi. Zaradi tega so danes najpogostejše uporabljane velikosti podslik 4x4, 8x8 in 16x16 slikovnih elementov. Standard JPEG naprimer predvideva velikost 8x8 slikovnih elementov.

### **5.2.1 Diskretna kosinusna transformacija**

Diskretna kosinusna transformacija (angl. Discrete Cosine Transformation -DCT) je danes najbolj razširjena transformacija za zgoščevanje slik. Služila nam bo kot primer, na katerem bomo razjasnili osnovno idejo delovanja transformacijskega kodiranja.

Večina današnjih transformacijskih postopkov vrši preslikavo signala v frekvenčni prostor. Vsi transformi te vrste so osnovani na Fourierovem teoremu. Po tem teoremu se da vsako zvezno periodično funkcijo zapisati kot vsoto sinusnih signalov različnih frekvenc z ustreznimi amplitudami in fazami, čemur pravimo fourierova transformacija (angl. Fourier Transformation - FT). Pravimo, da smo funkcijo preslikali v frekvenčni prostor. Kadar funkcija ni zvezna, temveč diskretna v času, lahko vzorce predstavimo z diskretno fourierovo transformacijo (ang. Discrete Fourier Transformation - DFT). Kot bomo videli predstavlja diskretna kosinusna transformacija le modifikacijo splošne fourierove transformacije.

#### **5.2.1.1 Osnovna ideja diskretne kosinusne transformacije**

V tem poglavju si bomo ogledali osnove diskretne kosinusne transformacije, ki bodo prikazane na primeru funkcije z eno samo neodvisno spremenljivko. Slika je v splošnem funkcija dveh neodvisnih spremenljivk, poleg tega le te predstavljajo razdaljo (dimenijo) in ne čas, kot je to običajno pri analizi električnih signalov. Vendar z matematičnega zornega kota te lastnosti na izpeljavo ne vplivajo.

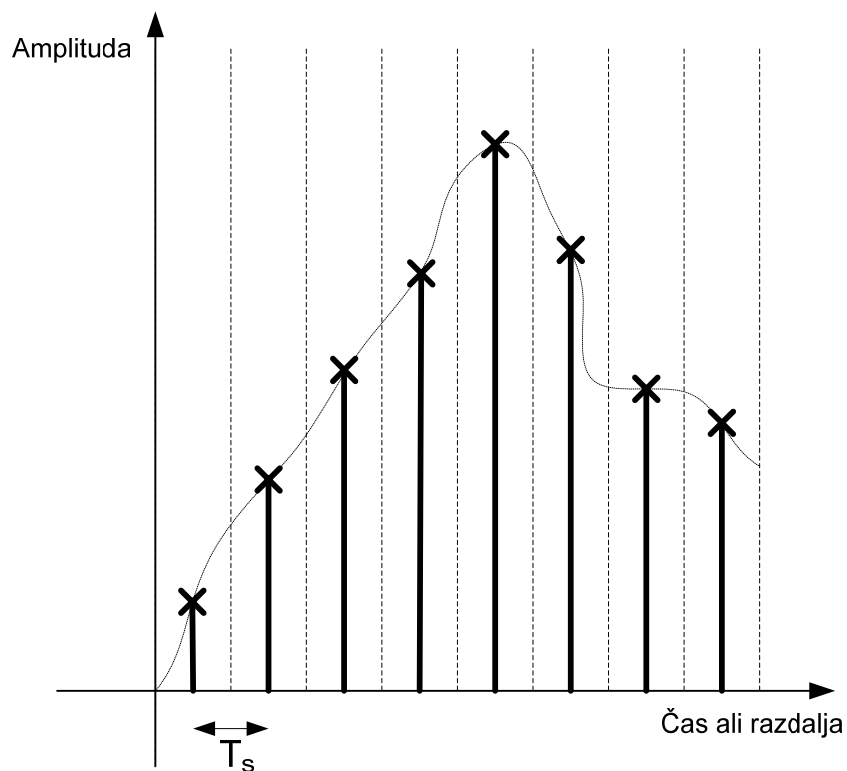
V splošnem po fourierovi analizi transformacijo vršimo nad zveznimi periodičnimi funkcijami. Slikovni signali ponavadi niso periodični. Poleg tega so slikovni signali, z katerimi se ukvarjamo v tej seminarski nalogi diskretni. Predstavljeni so namreč z vrednostmi posameznih slikovnih elementov v odvisnosti od dimenzije (lokacije elementa v

sliki). Preslikavo aperiodičnih diskretnih signalov v frekvenčni prostor izvedemo z pomočjo diskretne fourierove transformacije, ki jo bom opisal v nadaljevanju.

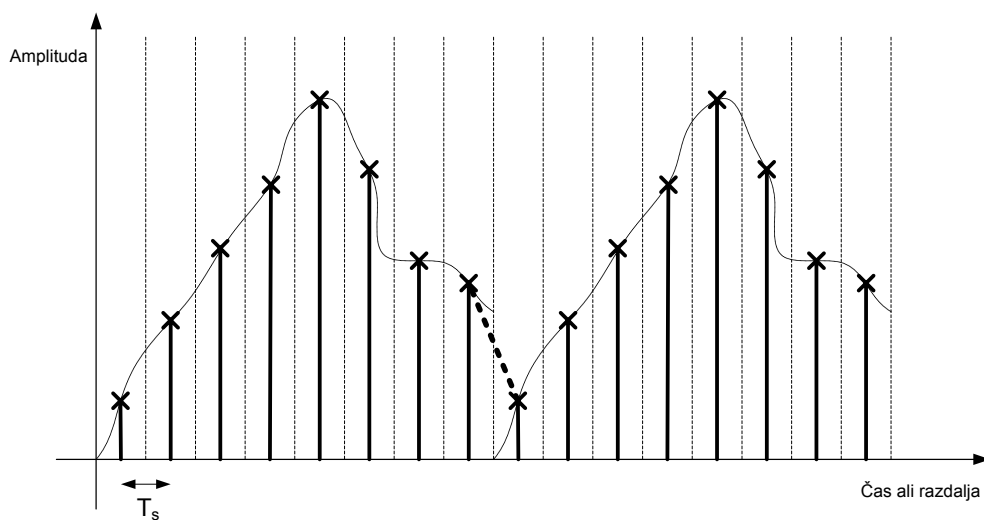
Na tem mestu je potrebno opozoriti, da je pri vzorčenju slikovnega signala, prav tako, kot v primeru signalov odvisnih od časa, potrebno upoštevati Niquistov teorem o vzorčenju. Ta pravi, da mora biti najvišja frekvenčna komponenta, ki se nahaja v signalu po frekvenci kvečjemu enaka polovici vzorčevalne frekvence. V naprotnem primeru pride do prekrivanja spektra in vrednosti koeficientov, ki jih izračunamo po diskretni fourierovi transformaciji niso pravilne. Zgodi se namreč, da se višje frekvenčne komponente (večje od polovice vzorčevalne frekvence) preslikajo v nižje frekvenčne komponente in nam tako popačijo osnovni frekvenčni pas. Zato je potrebno vse frekvenčne komponente, ki so višje od polovice vzorčevalne, iz slike izsejati še pred vzorčenjem.

Posledica vzorčnega teorema je, da v primeru diskretne fourierove transformacije računamo le koeficiente, ki predstavljajo frekvenčne komponente pri frekvencah, ki so manjše od  $f = 1/2 \cdot d$ , kjer nam  $d$  predstavlja dimenzijo slikovnega elementa. Ta izraz dobomo po analogiji z vzorčenjem časovnih funkcij. V primeru signala odvisnega od časa, bi bili vzorci pri vzorčenju z frekvenco  $f_s = 1/T_s$  razmaknjeni za čas  $T_s$ . Največja frekvenčna komponenta signala, ki jo še lahko pravilno predstavimo z takimi vzorci je  $f_s = 2f_{\max}$ , torej  $f_{\max} = f_s/2 = 1/2T_s$ , kar pomeni frekvenco z periodo  $2T_s$ . V krajevnem prostoru nam čas nadomesti dimenzija, torej časovna perioda  $2T_s$  preide v dimenzijsko periodo  $2 \cdot d[\text{cm}]$ , kjer nam  $d$  predstavlja velikost enega slikovnega elementa.

Kot primer diskretne fourierove transformacije nad funkcijo z eno samo neodvisno spremenljivko (čas) vzemimo zaporedje osmih vzorcev, odvisnih od časa in vzorčenih z frekvenco  $f_s$ . Iz te neperiodične funkcije napravimo periodično, in sicer tako, da po osmih vzorcih celotno sekvenco vzorcev ponovimo (Slika 5: Neperiodična funkcija predstavljena z osmimi vzorci in Slika 6).



Slika 5: Neperiodična funkcija predstavljena z osmimi vzorci



Slika 6: Predstavitev neperiodične funkcije z periodično – prvi način.

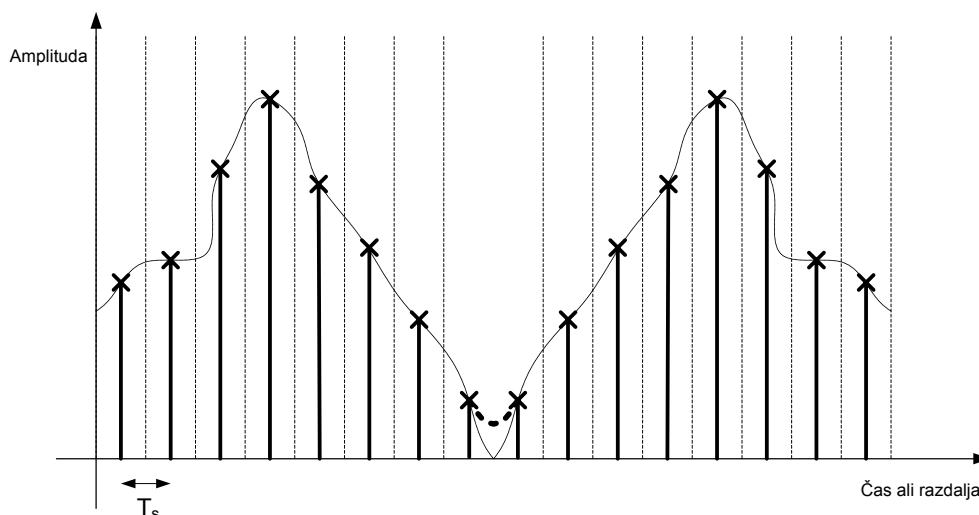
Predstavitev neperiodične funkcije z periodično z stališča rekonstrukcije signala ne predstavlja težav – pri obratni transformaciji pač vzamemo samo prvih 8 vzorcev. Dobili



smo torej periodično diskretno funkcijo z vzorci razmaknjenimi z čas  $T_s$ . To funkcijo sedaj preslikajmo v frekvenčni prostor. Diskretna fourierova transformacija predstavlja računanje koeficiente pri frekvancah  $f_s/8$ ,  $2f_s/8$ ,  $3f_s/8$  in  $4f_s/8$ . Pri tem je kot posledica Niquistovega teorema o vorčenju najvišja frekvenčna komponenta enaka  $f_{\max}=f_s/2$ . Začetnih osem vzorcev torej lahko predstavimo z štirimi frekvenčnimi komponentami (sinusoidami različnih frekvenc). Vsaka sinusoida je podana z amplitudo in fazo, torej so vrednosti koeficientov kompleksne. Začetnih osem ovzorcev lahko tretje v frekvenčnem prostoru opišemo z osmimi realnimi vrednostmi.

Predstavitve neperiodične funkcije z periodično ima seveda pomanjkljivost. Nekateri od izračunanih koeficientov transformacije bodo namreč odražali nenadno spremembo signala med »umetno ustvarjenimi« periodami (na Slika 6 označeno črtkano). Kadar je ta sprememba velika, bo precejšnja energija koeficientov predstavljala prav to, v originalnem signalu neobstoječo spremembo.

Neperiodično funkcijo lahko v periodično spremenimo tudi na drug način, kot ga kaže Slika 7. V tem primeru bo ena perioda sestavljena iz 16 vzorcev, v frekvenčnem prostoru bo signal torej predstavljen z koeficienti pri frekvancah  $f_s/16$ ,  $2f_s/16$ ,  $3f_s/16$ ,  $4f_s/16$ ,  $5f_s/16$ ,  $6f_s/16$ ,  $7f_s/16$  in  $8f_s/16$ . Za predstavitev v frekvenčnem prostoru bo torej potrebno izračunati 8 fourierovih koeficientov.



Slika 7: Predstavitev neperiodične funkcije z periodično – drugi način.

Z takšno predstavitevijo prvotno neperiodične funkcije smo izločili neželjeno spremembo signala na meji med periodami. Dobljeni fourierovi koeficienti sedaj v splošnem ne bodo več kompleksni temveč realni. To sledi iz dejstva, da je dobljena periodična funkcija simetrična glede na ordinatno os. Če kot izhodišče izberemo točko med periodama, potem je tako dobljena simetrična funkcija lahko predstavljena le kot vsota kosinusnih funkcij različnih frekvenc ( $\sin(-x)$  je namreč enak  $-(\sin x)$ , kar ne predstavlja na ordinato simetrične funkcije). To pomeni, da so fourierovi koeficienti realni in ne kompleksni. Tak način predstavitve neperiodične funkcije z periodično zahteva torej predstavitev signala z še enkrat več frekvenčnimi komponentami, vendar so koeficienti pri teh komponentah realni in za predstavitev signala tako zopet potrebujemo samo 8 realnih vrednosti. Opisani postopek predstavlja poenostavljeno razlago diskretne kosinusne transformacije.

#### **5.2.1.2 Diskretna kosinusna transformacija slikovnega signala**

Pri preslikavi določene funkcije v frekvenčni prostor je popolnoma nepomembno, ali je funkcija odvisna od časa ali dimenzije. Matematično gledano je namreč vseeno, ali nam spremenljivka predstavlja čas, dimenzijo ali kaj drugega. Tako kot lahko periodično časovno funkcijo predstavimo kot vsoto sinusnih valov različnih »časovnih« frekvenc (število nihajev na časovno enoto), lahko tudi krajevno funkcijo (odvisnost od razdalje do izhodišča) predstavimo kot vsoto različnih »krajevnih« frekvenc (število nihajev intenzitete na razdaljo).

Tudi odvisnost funkcije od večih neodvisnih spremenljivk pri transformacijah v frekvenčni prostor ne predstavlja problema. Kadar je funkcija odvisna od večih spremenljivk jo lahko zapišemo z koeficienti transformacije za vsako spremenljivko posebej. Intenziteta slikovnih elementov v sliki je dvodimenzionalna funkcija, pri kateri sta neodvisni spremenljivki oddaljenost od koordinatnega izhodišča v smereh  $x$  in  $y$ . Koeficiente transformacije torej sedaj računamo po dveh dimenzijah. Sliko, ki predstavlja dvodimenzionalno funkcijo v odvisnosti od razdalje od izhodišča tako opišemo z koeficienti v frekvenčnem prostoru dimenzijskih frekvenc.

Omenili smo že, da se transformacije ponavadi ne izvajajo nad celo sliko, pač pa nad bloki (podslikami) dimenzije  $N \times N$  slikovnih elementov, v katere se razdeli slika. Za blok dimenzije  $8 \times 8$  slikovnih elementov se diskretna kosinusna transformacija zapiše kot:

$$F(u, v) = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (\text{Enačba 2})$$

kjer se vrednosti  $C_u$  in  $C_v$  izračunajo:

$$\left\{ \begin{array}{ll} C_u = \frac{1}{\sqrt{2}} & \text{za } u = 0; \quad C_u = 1 \text{ drugje} \\ C_v = \frac{1}{\sqrt{2}} & \text{za } v = 0; \quad C_v = 1 \text{ drugje} \end{array} \right\}$$

Inverzno transformacijo dobimo po enačbi :

$$f(x, y) = \frac{1}{4} C_u C_v \sum_{u=0}^7 \sum_{v=0}^7 F(u, v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (\text{Enačba 3})$$

V obeh gornjih enačbah nam  $u$  in  $v$  predstavljata indkse koeficientov diskretne kosinusne transformacije in tečeta od 0 do 7. Koeiciente torej izračunamo pri vseh možnih kombinacijah  $(u, v)$ . Na ta način kot rezultat transformacije dobimo matriko 8x8 koeficientov. Za dano podsliko sestavljeno iz 64 slikovnih elementov torej dobimo 64 koeficientov transformacije.

Koeficient transformacije, ki se nahaja v levem zgornjem kotu matrike koeficientov naprimer

$$\text{izračunamo tako, da v enačbo } F(u, v) = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

(Enačba 2) vstavimo  $u=v=0$ . Dobimo vrednost:

$$F(0,0) = \frac{1}{8} \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \quad (\text{Enačba 4})$$

Če v gornji enačbi zanemarimo konstanto opazimo, da nam ta koeficient predstavlja povprečno vrednost vseh 64 slikovnih elementov podslike. Torej nam koeficient, ki se nahaja v levem zgornjem kotu matrike koeficientov predstavlja enosmerno komponento (frekvenco 0 v horizontalni in vertikalni smeri) podslike, nad katero smo transformacijo izvedli. Če se v matriki koeficientov pomikamo v desno nam koeficienti predstavljajo harmonske frekvence v horizontalni smeri slike. Če pa se pomikamo navzdol, nam koeficienti predstavljajo harmonske frekvence v vertikalni smeri slike. Koeficient v desnem spodnjem kotu predstavlja najvišji možni frekvenčni komponenti v horizontalni in vertikalni smeri.

Iz enačbe za inverzni diskretni kosinusni transform

$$f(x,y) = \frac{1}{4} C_u C_v \sum_{u=0}^7 \sum_{v=0}^7 F(u,v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (\text{Enačba 3})$$

lahko razberemo, da diskretna kosinusna transformacija slikovni signal predstavi kot ustrezno uteženo (z vrednostmi koeficientov) vsoto baznih funkcij, ki imajo kosinusni potek. V primeru, da bi pri preslikavi dobili samo en koeficient različen od nič (naprimer za

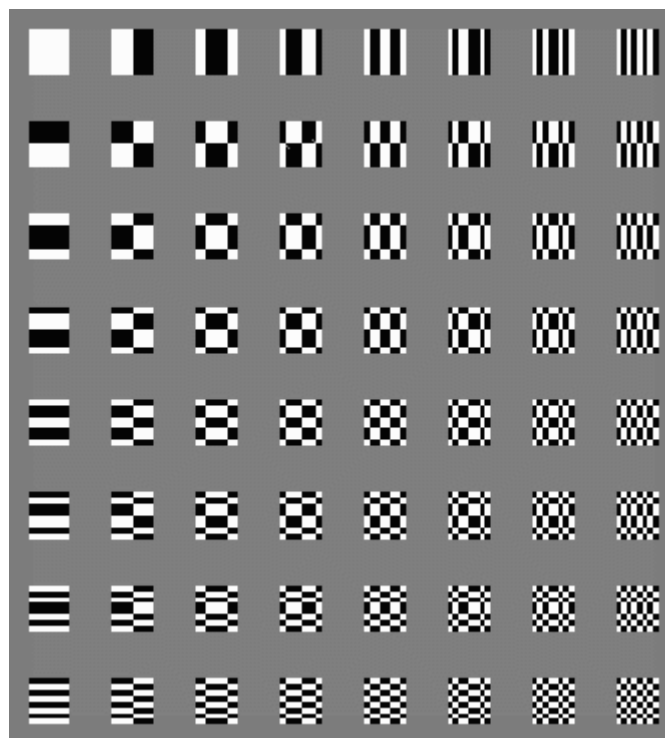
$$u=3, v=2), \text{ potem je po } f(x,y) = \frac{1}{4} C_u C_v \sum_{u=0}^7 \sum_{v=0}^7 F(u,v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

(Enačba 3) podslika, ki smo jo transformirali enaka:

$$f(x,y) = CONST. \cos\left(\frac{(2x+1)3\pi}{16}\right) \cos\left(\frac{(2y+1)2\pi}{16}\right) \quad (\text{Enačba 5})$$

Taka slika je prikazana v tretjem stolpcu in drugi vrstici na Slika 8. Vsakemu izmed 64 fourierovih koeficientov torej ustreza določena bazna slika. Če se v Slika 8 pomikamo v desno vidimo, da prostorska frekvenca narašča v horizontalni smeri, ko pa se pomikamo navzdol, pa baznim slikam prostorska frekvenca narašča v vertikalni smeri.

Poljubno slika se torej predstavi kot vsoto baznih slik, ki ustrezajo koeficientom ki so za dano sliko različni od nič. Diskretna kosinusna transformacija torej pomeni postopek iskanja matrike vrednosti koeficientov, inverzni transform pa te vrednosti zmnoži z odgovarjajočimi baznimi slikami in vse zmnožke sešteje – na ta način dobimo rekonstruirano sliko.



*Slika 8: Prikaz baznih slik za primer diskretne kosinusne transformacije nad bloki dimenzije 8x8 slikovnih elementov.*

Na tem mestu še enkrat poudatimo, da diskretna kosinusna transformacija deluje dobro samo v primeru slik vzorčenih v skladu z Nyquistovim teoremom. V nasprotnem primeru pride do prekrivanja spektrov in višje frekvenčne komponente nam bodo popačile koeficiente, ki predstavljajo nižje frekvenčne komponente. Pri rekonstrukciji slike bomo tako dobili velika popačenja.

### **5.2.2 Druge pogosto uporabljene transformacije**

Poleg diskretne kosinusne transformacije so se za namene zgoščevanja slik uveljavile tudi druge transformacije. Za zgoščevanje slik so tako še posebej zanimivi diskretni fourierov transform (DFT), diskretni hadamardov transform (DHT), diskretni walsh-ev transform (DWT), transformacija »Karhunen-Loeve« in druge. Transformacija »Karhunen-Loeve« je

transformacija, ki najbolj zgosti energijo v majhnem številu koeficientov, poleg tega pa tudi minimizira srednjo kvadratično vrednost napake, ki nastane zaradi kasnejše kvantizacije koeficientov. Tako je ta transformacija z stališča zgoščevanja optimalna. Problem je le v tem, da je ta transformacija računsko zelo kompleksna in se zato pri zgoščevanju slik zaenkrat še ni močno uveljavila. Ostale zgoraj naštetе transformacije so z stališča zgoščevanje slabše od transformacije »Karhunen-Loeve«. Od njih je najbolj učinkovita prav diskretna kosinusna transformacija, ki je zato trenutno najbolj uveljavljena transformacija na področju zgoščevanja slik.

V zadnjem desetletju se je pojavilo kar nekaj aplikacij za zgoščevanje slikovnih signalov, ki temeljijo na teoriji valjčkov (angl. Wavelet Theory). Z valjčno transformacijo (angl. Wavelet Transform), ki temelji na tej teoriji, lahko dosežemo veliko učinkovitost zgoščevanja. Obravnava vseh zgoraj naštetih transformov presega obseg te seminarske naloge, več informacij o njih lahko bralec dobi v literaturi [1].

## 6 Kvantizacija

Kvantizacija je postopek, ki dani vhodni vrednosti predpiše eno izmed nabora dovoljenih diskretnih izhodnih vrednosti (Slika 9). Izhodne vrednosti torej v večini primerov odstopajo od vhodnih, kar nam predstavlja kvantizacijski šum. Čimmanjše bo število možnih izhodnih vrednosti (nivojev kvantizatorja) večji bo kvantizacijski šum, vendar bo za zapis izhodne vrednosti potrebno manjše število bitov. Več kot ima kvantizator nivojev, več bitov bo potrebnih za predstavitev posameznega nivoja. Vendar to še ne pomeni, da bo po entropijskem kodiranju kvantiziranih vrednosti zaradi tega narasla tudi količina zgoščenih podatkov. Uporaba večjega števila bitov za zapis izhodnih vrednosti namreč pomeni manjši kvantizacijski šum in zaradi tega je lahko entropijsko kodiranje bolj učinkovito. Kvantizacija je lahko enakomerna ali neenakomerna. Pri enakomerni kvantizaciji so posamezni nivoji kvantizatorja med seboj enako oddaljeni. Pri neenakomirni kvantizaciji so si nekateri odločitveni nivoji kvantizatorja blizu (drobno kvantizacija), nekateri pa zelo narazen (groba kvantizacija). Območje vhodnega signala, ki je za predstavitev signala pomembno se kvantizira na drobno, za manj pomembna območja signala pa zadostuje groba kvantizacija. Iz povedanega sledi, da je izbira kvantizacijskega postopka oz. števila in porazdelitve nivojev za zgoščevanje signala zelo pomembna.

Kvantizacijo v procesu zgoščevanja slikovnih signalov lahko uporabimo na več načinov. Tako lahko surovi slikovni signal samo kvantiziramo in z tem že dobimo nek faktor zgoščevanja, ki pa ponavadi ne zadostuje. Bolj običajno se kvantizacija vrši po preoblikovanju surovega signala. Signal torej najprej transformiramo v obliko, ki je primernejša za kvantiziranje in entropijsko kodiranje. Dober primer, kako transformacija signala vpliva na izbiro kvantiziranja je naprimer prediktivno kodiranje (Poglavje 5.1). Na izhodu prediktorja namreč dobimo signal, ki bo z veliko verjetnostjo zavzemal majhne vrednosti, velike vrednosti izhodnega signala bodo zelo redke. V takem primeru je za kvantizacijo torej smiselno vzeti neenakomerni kvantizator. Ta bo vršil drobno kvantizacijo v območju malih vrednosti signala, v območju velikih vrednosti pa je kvantizacija lahko groba, saj signal velike vrednosti zavzame zelo redko in tako posledice grobe kvantizacije ne bodo tako opazne. Z sistemom, ki vrši predikcijo in nato kvantizacijo že dosegamo veliko boljše faktorje zgoščevanja. Še boljše faktorje pa dosegamo z transformacijskim kodiranjem, kjer po transformaciji kvantiziramo koeficiente, ki predstavljajo vhodni signal v domeni transformacije. Cena, ki jo plačamo se odraža v kompleksnosti sistema.

Zavedati se je potrebno, da pri kvantizaciji pride do izgube informacije, kar pomeni, da originalnega signala ne moremo v celoti rekonstruirati. Zgoščevalni postopki, ki uporabljajo kvantizacijo so torej izgubni. Pri izbiri kvantizatorja je torej potrebno paziti na to, da z kvantiziranjem izgubimo samo najmanj pomembno informacijo signala. Kaj nam predstavlja nepomembno informacijo, pa je seveda odvisno od aplikacije.

## **6.1 Vrste kvantizacije**

### **6.1.1 Skalarna kvantizacija**

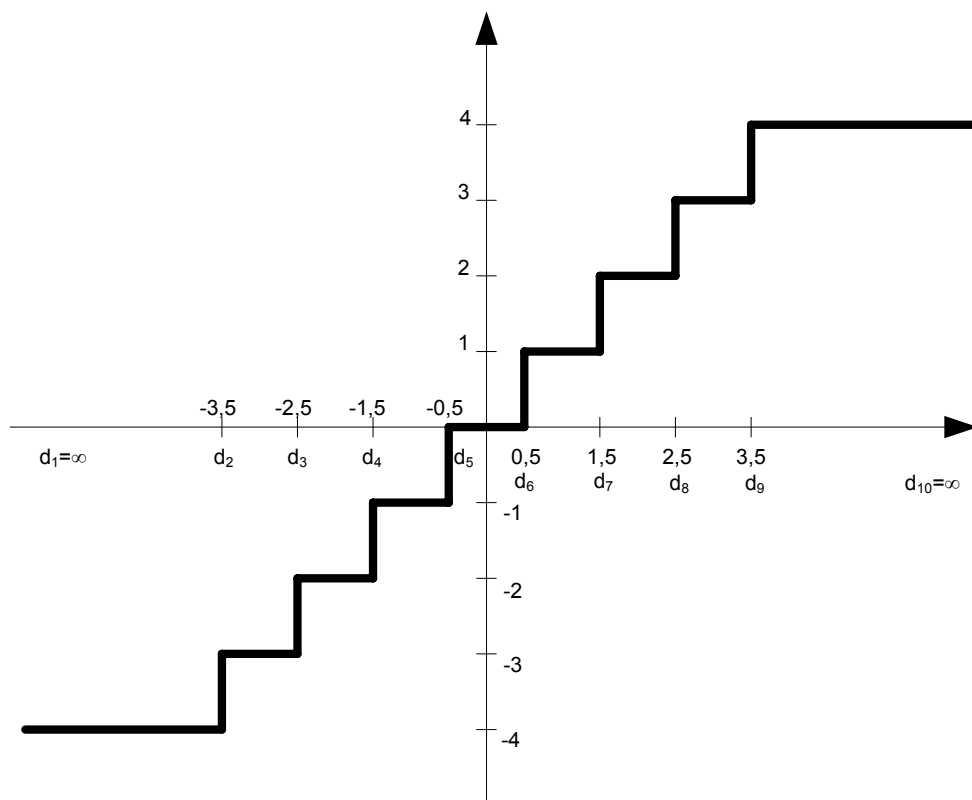
Skalarni kvantizatorji ocenjujejo vrednosti enega samega vhodnega parametra, naprimer intenzitete slikovenga elementa. Tak kvantizator si lahko predstavljamo kot funkcijo v obliki stopnic (Slika 9). Čimvečji je interval med stopnicami, tembolj groba je kvantizacija. Intervali med posameznimi stopnicami so lahko vsi enaki, v tem primeru govorimo o enakomerni kvantizaciji. Lahko pa so intervali med odločitvenimi nivoji neenakomerni in govorimo o neenakomerni kvantizaciji.

Kvantizator nam vhodne vrednosti signala (ki so lahko zvezne ali pa diskretne po amplitudi) preslika v izhodne simbole, pri čemer je število možnih izhodnih simbolov določeno z kvantizatorjem. Izhod kvantizatorja so torej simboli, ki jih ponavadi zapišemo kot posamezne kodne besede (nize bitov). Tako naprimer 8 možnih izhodnih simbolov na izhodu zakodiramo z po tremi biti. Takemu načinu zapisa izhodnih simbolov pravimo pulzno kodna modulacija (angl. Pulse Code Modulation - PCM).

#### **6.1.1.1 Enakomerni kvantizator**

Je najenostavnejši med kvantizatorji. Vsi odločitveni nivoji so med seboj enako razmaknjeni. Izkaže se, da pri rekonstrukciji signala najmanjšo srednjo kvadratično napako (angl. Mean Square Error) da primer, ko je vrednost izhodnega simbola kar enaka vsoti sosednjih odločitvenih nivojev deljeno z 2. Tako na sliki Slika 9 vidimo, da je izhod kvantizatorja za katerokoli vhodno vrednost med 0,5 in 1,5 enak 1. Enakomenremu kvantizatorju, ki minimizira srednjo kvadratično napako pravimo tudi optimalni enakomerni kvantizator.





Slika 9: Vhodno-izhodna funkcija enakomernega kvantizatorja.

Enakomerni kvantizatorji so primerni za kvantizacijo signalov, pri katerih je verjetnostna porazdelitvena funkcija približno ravna, kar pomeni, da vse vhodne vrednosti nastopajo z približno enako verjetnostjo.

#### 6.1.1.2 Neenakomerna kvantizacija

V primeru realne slike verjetnostna porazdelitvena funkcija intenzitete slikovnih elementov seveda ni ravna. Še manj to drži v primeru, da smo pred kvantizacijo naredili transformacijo vhodnega signala. Kadar tak izvor kvantiziramo z enakomernim kvantizatorjem ne dobimo najmanjše možne srednje kvadratične napake. Boljšo kvantizacijo namreč dosežemo, če vrednosti, ki nastopajo pogosto kvantiziramo natančno, tiste ki pa nastopajo redko pa kvantiziramo grobo. Neenakomernemu kvantizatorju, ki za podano število možnih izhodnih vrednosti naredi minimalno srednjo kvadratično napako pri dani verjetnostni porazdelitvi vhodne veličine rečemo Lloyd-Max kvantizator. To je optimalni neenakomerni kvantizator.

Potrebno pa je povedati, da majhna srednja kvadratična napaka še ne pomeni, da je izhodni signal iz kvantizatorja tudi primeren za dobro entropijsko kodiranje. Optimalni

neenakomerni kvantizator namreč prerazporedi odločitvene nivoje tako, da izenači verjetnosti, da bo vhodni signal padel v posamezna območja med odločitvenimi nivoji. To pomeni, da bodo vsi izhodni simboli nastopali z enako verjetnostjo. V primeru kvantizatorja z osmimi možnimi izhodnimi simboli bo entropija (Poglavje 7) signala na izhodu takšnega kvantizatorja enaka 3 bite na simbol, in vemo, da se pod to mejo z entropijskim kodiranjem ne moremo spustiti (teorem o izvornem kodiranju). Kot alternativo, bi lahko vzeli 16-nivojski enakomerni kvantizator, ki bi imel enako srednjo kvadratično napako kot jo je imel 8-nivojski neenakomerni kvantizator. Vsak izhodni simbol je sedaj predstavljen z štirimi biti vendar nekateri simboli nastopajo pogostejše kot drugi, torej je entropija izhodnega signala manjša od 4 bite na simbol, morda celo manjša od 3 bite na simbol. Z dobrim entropijskim kodirnikom, ki sledi kvantizatorju bi lahko za zapis takega signala potrebovali manj kot 3 bite na simbol. To je dokaz, da za čimboljše zgoščevanje ne potrebujemo optimalnega Lloyd-Max-ovega kvantizatorja. Poiskati moramo torej kvantizator, ki minimizira popačenje pri neki podani entropiji izhodnega signala. Takemu kvantizatorju pravimo entropijsko določen kvantizator (angl: Entropy Constrained Quantizer - ECQ). Teoretično je dokazano, da je entropijsko določen kvantizator za izvor brez spomina kar enakomerni kvantizator. Experimentalno je bilo tudi pokazano, da za ostale izvore enakomerni kvantizator dosega zmoglosti, ki so precej blizu z stališča kodiranja optimalnemu, entropijsko določenemu kvantizatorju. Iz tega sledi, da je za zgoščevanje v večini primerov najboljša kombinacija enakomernega kvantizatorja, ki mu sledi entropijski kodirnik.

#### **6.1.1.3 Adaptivna kvantizacija**

Med obravnavo neenakomerne kvantizacije smo že omenili, da imajo za različne verjetnostne porazdelitve vhodne naključne spremenljivke optimalni kvantizatorji različne porazdelitve odločitvenih nivojev. Kadar se signal spreminja z časom se z časom spreminja tudi njegova verjetnostna porazdelitev, to pa pomeni, da nek kvantizator za tak signal ne bo ves čas optimalen. Pri adaptivnem kvantizatorju se zato karakteristike kvantizatorja spreminjajo glede na statistiko vhodnega signala. Odločitveni nivoje se spreminjajo glede na trenutno funkcijo porazdelitve gostote verjetnosti, varianco in srednjo vrednost vhodnega signala.

### 6.1.2 Vektorska kvantizacija

Skalarna kvantizacija se vrši nad eno-dimenzijsko vhodno naključno spremenljivko. V primeru barvne slike, bi za kvantizacijo potrebovali torej tri skalarne kvantizatorje.

Vektorski kvantizator za razliko od skalarnega deluje nad več-dimenzijskim vhodnim signalom. Vhodne večdimenzijske vrednosti preslika v nek nabor rekonstrukcijskih vektorjev. Le ti so analogni rekonstrukcijskim nivojem pri skalarnem kvantizatorju, razlika pa je v tem, da so to večdimenzionalne vrednosti (naprimer nabor možnih izhodnih RGB vrednosti). Vsakemu rekonstrukcijskemu vektorju predpišemo neko število, ki mu pravimo indeks. Tako lahko naprimer številka 39 pomeni vektor z komponentami R=14, G=33 in B=57. Naboru kodnih števil skupaj z rekonstrukcijskimi vektroji pravimo kodna knjiga (angl. Codebook).

Vektorska kvantizacija poteka na način, da vsak vhodni vektor primerjamo z vsemi vektorji iz kodne knjige, izberemo pa tistega, ki da najmanjše odstopanje od originalne vrednosti. Vektorska kvantizacija je pri zgoščevanju zelo učinkovita, z njo lahko dosežemo rezultate, ki so zelo blizu teoretične meje. Slaba lastnost je v tem, da je to računsko zelo zahteven postopek. Za vsak signal z določeno statistiko moramo namreč najprej generirati kodno knjigo, pa tudi ko kodno knjigo poznamo je primerjanje vhodnih vrednosti z vsemi možnimi vektorji računsko zahtevna naloga.

Primer dobro poznanega standarda za zapis slik, ki uporablja vektorsko kvantizacijo je format GIF. Kot vhodni signal GIF algoritem vzame sliko v surovem formatu RGB. Algoritem analizira dano sliko in izbere 265 za zapis najprimernejših RGB vektorjev. Vsako 24 bitno RGB vrednost surovega formata nato nadomesti z 8 bitnim indeksom ustreznega vektorja. Na ta način še pred entropijskim kodiranjem dosežemo zgoščevanje v razmerju 1:3. Podan opis algoritma je seveda precej poenostavljen. GIF zapis vsebuje še cel kup postopkov za boljšo rekonstrukcijo slike. Eden od teh je naprimer drhtenje barve (angl. Color Dithering), pri katerem kodirnik za slikovne elemente iste barve po nekem ključu dodeli različne vektorje, ki pa so vsi blizu barve originalnega slikovnega elementa. Ker človeško oko ne vidi posameznih slikovnih elementov, pač pa celoto, kot skupni efekt dobimo barvo, ki bolj odgovarja originalni barvi slikovnega elementa, kot pa posamezni dodeljeni vektorji.

## 7 Entropijsko kodiranje simbolov

Izhod iz kvantizatorja v zgoščevalnem sistemu predstavljajo nizi bitov. Kadar vse izhodne simbole iz kvantizatorja kodiramo z istim številom bitov, bo v primeru neenakomirne verjetnostne porazdelitve simbolov tak zapis vseboval veliko redundance. Z entropijskim kodiranjem izločamo ta tip redundanco.

Z uporabo različnih postopkov transformacije in kvantizacijo, sliko spremenimo v zaporedje simbolov. Za uspešen prenos in shranjevanje moramo zaporedje simbolov pretvoriti v bitno zaporedje. Število bitov, ki jih potrebujemo za zapis posameznega simbola, je odvisno od števila možnih simbolov, ki se pojavljajo v zaporedju. Najmanjše povprečno število bitov, ki jih potrebujemo za zapis simbolov nekega izvora z dano statistiko je seveda odvisno od količine informacije, ki jo signal vsebuje. Z teoretično obravnavo informacije izvora se ukvarja t.i. informacijska teorija. Kot začetnika informacijske teorije štejemo raziskovalca Claude Elwood Shannon-a. Leta 1948 je objavil študijo z naslovom "A Mathematical Theory of Communication", v kateri je postavil teoretične meje na področju informacijske teorije. Osnovo informacijske teorije predstavlja

enačba za izračun količine informacije, ki jo nek simbol nosi. Po tej enačbi ( $I_k = \log\left(\frac{1}{p_k}\right)$ )

(Enačba 6) je

informacija simbola obratno sorazmerna verjetnosti njegovega pojavljanja. Čimpogosteje se nek simbol v signalu pojavlja temmanj informacije nosi.

$$I_k = \log\left(\frac{1}{p_k}\right) \quad (\text{Enačba 6})$$

Naboru vseh možnih simbolov, ki nastopajo v signalu pravimo abeceda. Kadar vsi simboli abecede nastopajo z enako verjetnostjo in abecedo sestavlja  $n$  simbolov, se povprečno

informacijo, ki jo nosi en simbol izračuna po enačbi  $I = \log\left(\frac{1}{p}\right) = \log n$

(Enačba 7.)

$$I = \log\left(\frac{1}{p}\right) = \log n \quad (\text{Enačba 7})$$

Kadar simboli ne nastopajo z enako verjetnostjo, se povprečna informacija, ki jo nosi en

simbol izračuna po enačbi 
$$H = -\sum_{i=1}^n p(s_k) \log_2\left(\frac{1}{p(s_k)}\right)$$

(Enačba 8 .

$$H = -\sum_{i=1}^n p(s_k) \log_2\left(\frac{1}{p(s_k)}\right) \quad (\text{Enačba 8})$$

Tej veličini pravimo entropija zaporedja simbolov. Entropija je torej merilo, ki nam pove koliko informacije v povprečju nosi en simbol signala. Vrednost entropije signala je določeno z njegovo statistiko. Entropija nam predstavlja teoretično spodnjo mejo, ki govori o tem koliko informacijskih bitov potrebujemo za zapis enega simbola signala z določeno statistiko.

Simbole nekega signala ponavadi zakodiramo z nizom bitov. Za zapis vsakega simbola lahko uporabimo enako število bitov, bolj smiselno pa je, da za pogostejše simbole uporabimo krajšo kodno besedo, kot za simbole, ki nastopajo poredko. Teorem o izvornem kodiranju pravi, da se z takim načinom kodiranja lahko poljubno približamo vrednosti, ki je določena z entropijo izvora, nikakor pa te vrednosti ne moremo preseči. Tako za primer izvora z entropijo 4.17 informacijskih bitov na simbol v povprečju potrebujemo najmanj 4.17 bitov za zapis enega simbola. Seveda tudi ta vrednost v praksi ni dosegljiva, saj bi potrebovali optimalni entropijski kodirnik. Kljub temu, da teoretična meja, ki jo predstavlja entropija izvora v praksi ni dosegljiva, nam entropija predstavlja dobro merilo za vrednotenje in primerjavo različnih postopkov kodiranja simbolov.

Na tem mestu je potrebno poudariti, da vsa zgoraj opisana teoretična spoznanja veljajo za diskretni izvor brez spomina (angl. Discrete Memoryless Source - DMS). To je izvor, katerega zaporedni simboli lahko zavzamejo samo diskretne amplitude in so med seboj neodvisni, kar pomeni, da iz vrednosti predhodnjih simbolov ne moremo sklepati na vrednost naslednjega simbola.

### 2.3.1 Kodiranje simbolov z dolžino ponovitev

Najpreprostejši sistem, ki za skrajšanje zapisa upošteva statistiko ponovitev posameznih simbolov v zaporedju je takoimenovano kodiranje simbolov z dolžino ponovitev (angl. Runlength Coding). Kodirno zaporedje je paroma sestavljeno iz zaporedij simbolov in števila ponovitev posameznega simbola. Pri tem simbol zapišemo v njegovi izvorni abecedi z enakim številom bitov na simbol, za kodiranje števila ponovitev simbola pa vpeljemo nove simbole. Kodna zaporedja, uporabljena za kodiranje simbolov in njihovega števila ponovitev se morajo razlikovati. Na ta način za kodiranje več enakih zaporednih simbolov uporabimo le dva simbola (prvi pove za kateri simbol gre drugi pa število njegovih ponovitev). Postopek za kodiranje prikazuje tabela Tabela 1.

Niz simbolov	10	0	0	0	0	10	10	20	10	0	0	0
Simbol	10				0		10	20	10			0
Št. ponovitev	1				4		2	1	1			3

*Tabela 1: Postopek kodiranja z dolžino ponovitev*

Kodiranje simbolov z dolžino ponovitev v resnici ne predstavlja pravega kodirnega postopka, saj poskrbi le za drugačno obliko zapisa istega zaporedja. Takšen kodirni pristop je uspešen le v primeru, da v nizu nastopajo dolga zaporedja enakih simbolov. V primeru, v katerem ne prihaja do ponovitev simbolov, s takšnim pristopom celo podvojimo število simbolov, ki jih potrebujemo za zapis celotnega niza.

### 2.3.2 Huffmanovo kodiranje

Huffmanovo kodiranje temelji na postopki, pri katerem manj verjetne simbole kodiramo z daljšimi, bolj verjetne pa z krajšimi binarnimi zaporedji. Pri kodiranju abecede n različnih

simbolov  $s_k$ , ki nastopajo z verjetnostjo  $p(s_k)$  Huffmanov postopek skuša oblikovati bitne zapise simbolov  $c_k$ , katerih povprečno bitno dolžino izračunamo po enačbi

$$\bar{l}(c) = \sum_{k=1}^n p(s_k) l(c_k) \quad \text{Enačba 9.}$$

$$\bar{l}(c) = \sum_{k=1}^n p(s_k) l(c_k) \quad \text{Enačba 9}$$

Ker za posamezne simbole uporabimo bitne zapise različnih dolžin, moramo zagotoviti enolično dekodiranje zaporedja simbolov. Bitni zapisi morajo biti torej kodirani po načinu kodiranja z predpono. To je kodiranje, pri katerem nobeno zaporedje bitov, ki predstavljajo posamezen simbol ne sem imeti enake predpone kot katerikoli drug zapis v abecedi. Postopek za določitev takšnega zaporedja je odkril raziskovalec Huffman. Dolžina kode Huffmanovega zaporedja znaša vsaj 1 bit na simbol, kar predstavlja slabost v primerih, kjer je porazdelitev simbolov izvora izrazito neenakomerna. Poleg tega so kode spremenljive dolžine, katerih predstavnik je Huffmanova koda, neuporabne v primeru močnega spreminjanja statistike izvora. Takoj, ko se spremeni verjetnostna porazdelitev izvora, bi morali spremeniti tudi kodno tabelo, da bi bila koda učinkovita. Zato so kode spremenljive dolžine uporabne samo v primeru, da se statistika izvora bistveno ne spreminja. Kadar pa se statistika spreminja močno lahko te kode celo povečajo količino podatkov potrebno za zapis signala.

### 2.3.3 Aritmetično kodiranje

V primeru Huffmanovega kodiranja je vsak simbol zakodiran z kodno besedo, ki je sestavljena iz celoštevilčnega števila bitov. Tako nobena kodna beseda ne more biti krajša od enega bita in Huffmanovo kodiranje zato ni učinkovito v primeru vhodnih signalov z izrazito različno porazdelitvijo verjetnosti pri posameznih simbolih. Obstajajo sicer tehnike, ki to pomanjkljivost skušajo izboljšati na način, da simbole kombinirajo v večje bloke, njihova slabost pa je večja kompleksnost. Poleg tega je Huffmanovo kodiranje učinkovito le v primeru izvora, ki se mu statistika z časom ne spreminja.

Te pomanjkljivosti odpravlja družina entropijskih kodirnikov, ki jim pravimo aritmetični kodirniki. Z uporabo aritmetičnega kodiranja lahko za zapis uporabimo tudi manj kot 1 bit na posamezni simbol. Izboljšano razmerje zgoščevanja, ki ga na ta način dosežemo, je še posebej izrazito v primeru kodiranja dolgih zaporedij.

Aritmetično kodiranje je kodiranje z neprekinjenim bitnim tokom. Na vhod nam prihaja zaporedje simbolov in na izhodu dobimo zaporedje bitov, pri tem pa simboli niso enolično določeni z nizom bitov, ampak so odvisni tudi od prejšnjih simbolov (podobno kot pri konvolucijskem kodiranju). Kodiranje je osnovano na velikem številu binarnih odločitev in na neprestanem osveževanju ocenjevanja verjetnostne porazdelitve izvora. Kodiranje se tako neprestano prilagaja statistiki izvora. Aritmetično kodiranje je del standarda JPEG in predstavlja enega bolj učinkovitih načinov zgoščevanja.



## 8 Najpogostejši standardi za zapis mirujoče slike

Standardi za zapis slik ponavadi obsegajo več kot zgolj definicijo uporabljenih algoritmov za zgoščevanje zapisa. Nekateri standardi dopuščajo uporabo različnih tehnologij za zgoščevanje, ponavadi pa obstaja tudi več različnih standardov, ki uporabljajo enake ali vsaj sorodne tehnologije zgoščevanja. Osnovni namen vseh krovnih standardov je zagotoviti čimkvalitetnejši zapis slike z čimkrajšim bitnim zapisom.

### 8.1 Zapis JPEG

Med standardiziranimi postopki za zgoščevanje slik se danes največ uporabljajo izgubni algoritmi na podlagi razdelitve slike v bloke, nad katerimi se potem izvede transformacija, najpogosteje diskretna kosinusna transformacija. Takšen način zgoščevanja je specificiran tudi v standardu JPEG (Joint Photographic Experts Group). Zgoščevanje po standardu JPEG je v praksi široko razširjen način zgoščevanja slikovnega signala. Algoritem v primerjavi z drugimi rešitvami procesno ni zahteven. Uporablja se na področju zgoščevanja mirujočih slik, z modifikacijami pa tudi za zgoščevanje videa. Standard vključuje brezizgubne in izgubne postopke zgoščevanja. Podpira naslednje postopke:

- Sekvenčno na DCT osnovano zgoščevanje, ki je v praksi najbolj razširjen postopek;
- Napredujoče kodiranje, pri katerem dekodiranje slike poteka v večih korakih. Grob obris slike se tako lahko prikaže zelo hitro, nato pa se slika v nekaj korakih izostri;
- Brezizgubno zgoščevanje;
- Hierarhično kodiranje, ki podpira hkratni zapis slike v več ločljivostih.

Sekvenčno na DCT osnovano zgoščevanje temelji na uporabi diskretne kosinusne transformacije po blokih (velikost blokov je 8x8 slikovnih elementov), ki ji sledi kvatizacija koeficientov in Aritmetično ali Huffmanovo kodiranje. Po opravljeni transformaciji vsakega izmed 64 koeficientov dodatno utežimo. Uteži so izbrane glede na pomen posameznih spektralnih komponent za človeško zaznavanje. Pred nadaljnjo obdelavo koeficiente zaokrožimo na cele vrednosti. Ta način je primeren za večino aplikacij.

Napredujoče kodiranje se vse bolj uveljavlja na področju interaktivnih omrežnih storitev. Za razliko od sekvenčnega kodiranja se pri napredujočem postopku vsaka slikovna komponenta kodira v več korakih. Prva pretvorba daje slabo kvaliteto slike, vendar jo lahko

prikažemo ali prenesemo preko omrežja veliko hitreje kot sliko končne kvalitete. Prednost tega načina pride do izraza zlasti pri obravnavi slik visoke ločljivosti.

Brezizgubno zgoščevanje poteka na osnovi linearne predikcije. Razliko med napovedjo in dejansko vrednostjo v naslednji stopnji entropijsko kodiramo. V dekodirju na sprejemni strani upoštevamo izbrano metodo predikcije (standard namreč podpira več različnih modelov napovedovanja), z prediktorjem izračunanim vrednostim pa prištejemo zapisane razlike in na ta način dobimo rekonstruirane vrednosti slikovnih elementov. Količina podatkov za zapis slike in z tem tudi učinkovitost zgoščevanja je v tem primeru v večji meri odvisna od izbire modela za napovedovanje vrednosti slikovnih elementov.

V primeru hierarhičnega kodiranja se slika kodira v več ločljivostih. S tem načinom sliko visoke ločljivosti zapišemo kot zaporedje slik z različnimi ločljivostmi, od najnižje do najvišje. Ta način je dobrodošel predvsem pri prenosu slik. Na sprejemni strani sliko namreč gradimo na podlagi predhodnih slik nižje ločljivosti. Prednost postopka je torej v tem, da imamo na sprejemni strani sliko v nižji ločljivosti na voljo, še preden sprejmemo in dekodiramo sliko v polni ločljivosti (predogled).

## **8.2 Zapis JPEG 2000**

Pobuda za razvoj standarda JPEG 2000 je nastala v želji, da bi z uporabo učinkovitejših kodirnih postopkov izboljšali rezultate, ki jih dosegamo z uporabo standarda JPEG. Standard JPEG 2000 vključuje več različnih postopkov za zgoščevanje kot njegov predhodnik, te pa obsegajo tako izgubno kot tudi brezizgubno zgoščevanje. Standard med prvimi vključuje uporabo podpasovnega zgoščevanja (valjčna transformacija) namesto klasične uporabe diskretne kosinusne transformacije. S tem predlagatelji želijo vplivati predvsem na moteče blokovne popačitve, značilne za izgubni način zgoščevanja po standardu JPEG. Z uporabo valčne transformacije na celotnem področju slike se v primeru visoke stopnje zgoščevanja v resnici izognemo vidnim prehodom med bloki, pojavijo pa se druge napake, kot je na primer zvonjenje robov. Poleg razširitve nabora postopkov za zgoščevanje standard JPEG 2000 uvaja tudi področno zgoščevanje, podatke za barvno korekcijo in zmogljiv pristop z uporabo 256 kanalov za vključevanje metapodatkov.

### 8.3 Zapis TIFF

Standard TIFF (Tagged Image File Format) je eden od najbolj razširjenih in prilagodljivih formatov zapisa slik v javni lasti. Zapis je zelo prilagodljiv, neodvisen od strojne opreme, podpirajo pa ga proizvajalci širokega spektra periferne opreme. Zapis vsebuje podatke, potrebne za barvno korekcijo slike. V primeru velikih slik zapis TIFF omogoča delo s posameznimi pravokotnimi področji slike; tako se lahko izognemo obdelavi celotnih vrstic. Dovoljen je zapis slikovnih elementov v poljubni bitni ločljivosti, podpora barvnim prostorom pa vključuje širok spekter od črnobelih, sivinskih slik, indeksiranega barvnega prostora, RGB do krominančnih zapisov. Postopki zgoščevanja dopuščajo izgubno in brezizgubno zgoščevanje ter vključujejo zapise, kot so nezgoščeni podatki, zgoščevanje po algoritmu LZW (Lempel-Ziv-Welch), telefaksimilni zapis po CCITT G3 in G4 ter celo JPEG.

### 8.4 Zapis GIF

GIF (Graphics Interchange Format) je 8 bitni slikovni format, ki uporablja nabor 256 barv (podprostor 24 bitnega RGB modela). GIF uporablja brezizgubno zgoščevanje osnovano na LZW algoritmu, ta pa je bil do nedavnega patentiran in zato GIF ni predstavljal odprtokodne rešitev. Do izgub informacije seveda ne pride le v primeru, da vsebuje prvotna digitalna slika le 256 barvnih odtenkov. Format GIF torej ni zelo primeren za barvne fotografske slike, primernejši pa je za enostavnejše slike kot so grafike, logotipi itd. Uporablja se lahko tudi za zgoščevanje monokromatskih slik, vendar zgoščevalna tehnika takim slikam ni najbolj prilagojena, zato pri zgoščevanju takih monokromatskih slik ne dobimo tako velikih faktorjev zgoščevanja kot ob uporabi nekaterih drugih standardov.

### 8.5 Zapis PNG

Format zapisa PNG (Portable Network Graphics) je nastal kot odgovor na licenčno zaščito znanega formata GIF. Format zapisa PNG temelji na načelih "odprte kode" in dopušča prosto izmenjavo in nadaljni razvoj formata. Trenutno uradno podpira do 48-bitni barvni in do 16-bitni sivinski grafični zapis, kar je s stališča izrabe pasovne širine za prenos slik razmeroma potratno in navedene mejne vrednosti zato niso pogosto uporabljane. Zgoščevanje PNG sodi v kategorijo brezizgubnih kodirnih postopkov. Učinkovitost zgoščevanja naj bi bila višja od učinkovitosti postopkov GIF. Format zapisa vključuje

podatke za barvno korekcijo prikaza, nudi popolno podporo prozornosti grafičnih elementov in v prepletenem načinu kodiranja omogoča progresiven delni prikaz gradiva med prenosom podatkov. Kot večina sodobnih standardov vključuje podporo za vnos metapodatkov znotraj samega zapisa.

## 9 ZAKLJUČEK

Spominski prostor potreben za shranjevanje slikovnih signalov, še toliko bolj pa frekvenčni spekter, potreben za prenos takšnih signalov sta omejeni in zato dragi dobrini. To je v primeru slikovnega signala še toliko bolj pereče, saj slikovni signal vsebuje veliko količino informacije. Ob vsevečji težnji po čimvečji kvaliteti zmogljivosti strojne in programske opreme zaostajajo za zmogljivostmi, ki jih zahtevamo za shranjevanje in prenos slikovnih signalov. Potrebno je bilo razviti postopke, ki nam količino podatkov potrebno za zapis slik karseda zmanjšajo. Zmanjšanje količine podatkov za zapis se doseže z odstranjevanjem redundance in irelevance iz slikovnega signala. V pričujočem seminarskem delu sem zato predstavil različne načine odstranjevanja redundance in irelevance iz slikovnih signalov, podal njihove prednosti in slabosti in nakazal kateri načini se uporabljajo v najbolj razširjenih standardih za zapis mirujočih slik.

Postopke za zgoščevanje slikovnega signala v prvi vrsti delimo na izgubne in neizgubne. V primeru neizgubnega postopka lahko originalni signal v celoti rekonstruiramo. Te vrste postopki temeljijo zgolj na odstranjevanju redundance, ki jo je v slikah zapisanih v surovem formatu običajno veliko. Kljub veliki količini redundance neizgubno zgoščevanje ponavadi ni dovolj učinkovito. Za željeno stopnjo zgoščevanja je tako potrebno uporabiti tudi izgubne postopke. Izgube večinoma nastanejo v fazi kvantizacije in pri zaokroževanju in odmetavanju za rekonstrukcijo nepomembnih koeficientov po transformiranju signala.

Sistemov z brezizgubnim zgoščevanjem ni težko medsebojno primerjati. Boljši je namreč tisti, ki doseže večji faktor zgoščevanja. V primeru izgubnega zgoščevanja pa originalnega signala ni mogoče v celoti rekonstruirati. Nastalo napako je potrebno nek način ovrednotiti. Vrednotenje je lahko objektivno, kjer napako izračunamo po določeni enačbi, lahko pa je subjektivno, kjer stopnjo popačenja določi človek. Objektivna vrednotenja bi bila v idealnem primeru popolnoma korelirana z subjektivnimi, vendar takšnega postopka objektivnega vrednotenja danes še ni poznano. Tako redino pravo vrednotenje popačenja ostaja v obliki subjektivnih primerjalnih testov, z velikim številom opazovalcev, ter pod nadzorovanimi pogoji. Tak proces je seveda drag, počasen in nepraktičen zato je na področju zgoščevanja slikovnega signala ogromno raziskovalnega truda usmerjenega prav v iskanje objektivne metode vrednotenja kvalitete slike.

Glede na težnje k povečevanju faktorja zgoščevanja, in dejstvo, da postaja strojna in programska oprema vse zmoglivejša lahko zaključimo, da se bodo v prihodnje vsebolj uveljavili algoritmi za zgoščevanje z veliko računsko kompleksnostjo. Splošno razširjeni naj bi tako postali postopki, kot so vektorska kvantizacija, transformacije z valjčno preslikavo in drugi. Neglede na uporabljeni algoritem pa je jasno, da teoretične meje, ki jo je že pred več kot 50 leti postavil Shannon z brezizgubnimi algoritmi ni mogoče preeči. Pričakovati je torej pospešen razvoj algoritmov, ki iz slikovnega signala odstranjujejo irelevanco, pri tem pa ostane popačenje slikovenga signala čimmanjše.

## 10 UPORABLJENA LITERATURA

- [1] Yun Q. Shi, Huifang Sun (2000); *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms*; CRC Press LLC; 2000.
- [2] Tinku Acharya, Ping-Sing Tsai (2005); *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*; John Wiley & Sons, Inc.; 2005.
- [3] John W. Woods (2006); *Multidimensional Signal, Image, and Video Processing and Coding*; Academic Press; Elsevier Inc.; 2006.
- [4] Peter Symes (2001); *Video Compression Demystified*; McGraw-Hill Inc.; 2001.
- [5] Urban Burnik (2002); *Optimalno kodiranje za prenos slik po ozkopasovnih prenosnih medijih : doktorska disertacija*; Ljubljana 2002.