Univerzitet u Beogradu

Matematički fakultet

Vesna Vučković

Optimalna snaga žiga belog Gausovog šuma

Doktorska disertacija

Beograd

2010.

Univerzitet u Beogradu – Matematički fakultet

Doktorska disertacija

Autor: Vesna Vučković

Naslov disertacije: Optimalna snaga žiga belog Gausovog šuma

Mentor: prof. dr Žarko Mijajlović

Članovi komisije: prof. dr Miodrag Živković
prof. dr Milan Dražić
prof. dr Zoran Ognjanović

Uvod

Tema ovog rada je:

> **Naći optimalnu snagu ugradnje (digitalnog vodenog) žiga belog Gausovog šuma u sliku u nijansama sive boje.**
>
> Reč "*optimalna*" ovde znači: minimalna koja garantuje detektabilnost žiga posle očekivane modifikacije slike.

Modifikacija može biti kompresija sa gubicima, ali takođe i neko drugo valumetrijsko ili geometrijsko izobličenje slike (promena sjajnosti i kontrasta, zamućenje, izoštravanje, rotacija, opsecanje, promena dimenzije,...)

Ovaj rad nastao je u pokušaju da se takva *optimalna* snaga odredi za ugradnju žiga u sliku za koju se očekuje da će biti izložena kompresiji sa gubicima (na primer, JPEG ili DjVu). Zato uz praksu digitalnog vodenog žiga, i kompresija sa gubicima čini centralni deo rada. Kasnije se pojavila ideja da se ispita ponašanje žiga i prema drugim izobličenjima, pa je tome posvećeno poglavlje na kraju teksta.

Rad se sastoji iz tri dela. U prva dva dela (poglavlja od 1 do 8) ukratko se navode poznate činjenice, potrebne za razumevanje trećeg dela. Treći deo (poglavlja od 9 do 13) sadrži **originalne rezultate disertacije**.

Prvi deo, *Digitalizacija i kompresija slika*, sastoji se iz tri poglavlja.

U poglavlju 1 (*Digitalizacija slika*) uvode se neki osnovni pojmovi u vezi sa prikazom slike u računaru: rezolucija slike, dubina boje, fajl formati. Zatim sledi (**originalni**) predlog vizuelizacije proizvoljne matrice; predloženo rešenje biće korišćeno u radu tamo gde je potrebno predstaviti slikom proizvoljnu (realnu ili kompleksnu) matricu.

Poglavlje 2 uvodi neke pojmove u vezi sa *kompresijom podataka* (entropija, redundanca, kompresija sa i bez gubitaka). Navedene su neke često korišćene tehnike kompresije bez gubitaka (Hafmenovo, aritmetičko, RLE, rečničko kodiranje).

U poglavlju 3 govori se o *kompresiji sa gubicima* (JPEG kompresija, DCT, DFT, kompresija talasićima, kompresija fraktalima).

Drugi deo sadrži tekst koji treba da upozna čitaoca sa disciplinama skrivene komunikacije i *sakrivanja informacije u digitalne radove*, a pre svega sa praksom digitalnog vodenog žiga.
Poglavlje 4 (*Digitalni vodeni žig*) uvodi osnovne pojmove u vezi sa digitalnim vodenim žigom: razlozi pojave, aplikacije, i svojstva koje moraju posedovati.
U poglavlju 5 (*Praksa žiga, steganografija, kriptografija*) prvo se ove tri discipline porede među sobom. Zatim se navodi nekoliko interesantnih *steganografskih* tehnika.
U poglavlju 6 navode se *svojstva žiga*, bitna za ovaj rad (punjenje, efikasnost, robusnost, sigurnost, vernost).
Poglavlje 7 daje kratak pregled postojećih *robusnih tehnika digitalnog vodenog žiga*.
Poglavlje 8 uvodi čitaoca u algoritam (ugradnje i detekcije) *žiga belog Gausovog šuma*, o kome se govori u nastavku rada.

Deo 3 sadrži (**originalne**) rezultate istraživanja za *optimalnom snagom ugradnje* žiga belog Gausovog šuma u sliku u nijansama sive boje.
U poglavlju 9 određuje se optimalna snaga za *efikasnu ugradnju*. Izvodi se **egzaktna matematička formula**, bazirana na svojstvima normalne raspodele.
Poglavlje 10 bavi se *robusnošću žiga prema* očekivanoj *kompresiji* i predlaže postupak određivanja optimalne snage ugradnje, da bi žig i posle kompresije bio detektabilan.
Poglavlje 11 analizira *ugradnju žiga belog Gausovog šuma* preko cele slike *u domenu transformacije*. Pokazuje se da nema suštinske razlike između takve i ugradnje u prostornom domenu (optimalna snaga ugradnje je ista).
Poglavlje 12 govori o efikasnosti i robusnosti *ugradnje žiga belog Gausovog šuma u podslike u domenu transformacije*. Posebno se razmatra ugradnja u grupu ili neke pojedinačne potkanale 8×8 blokovske DCT.
Sadržaj poglavlja 13 (*Žig belog Gausovog šuma i druge modifikacije slike*) su neki drugi mogući napadi – izobličenja slike. Za takve napade se traži optimalna snaga ugradnje, da bi im žig odoleo. Posebno se analizira slučaj rotacije.
U poslednjem poglavlju (*Zaključci i dalji rad*) ukratko se sumiraju najvažniji dobijeni rezultati, i navode moguće oblasti istraživanja za budući rad.

Tema ove disertacije je multidisciplinarna. Mada ona pripada obradi slika, koja je pre svega računarska, u određivanju optimalne snage žiga koristi se u velikoj meri matematika (pre svega linearna algebra i teorija verovatnoće). U Dodatku (*Matematičke osnove*) dat je kratak prikaz matematičkih znanja potrebnih za razumevanje teksta (skalarni proizvod, norma, linearne i ortogonalne transformacije, normalna raspodela) i ukazano je na kontekst u kome su ta matematička znanja u ovom radu korišćena..

Rečnik stručnih termina sastoji se iz dva dela: srpsko-engleskog i englesko-srpskog. Tu su navedeni termini korišćeni u radu (veći deo tih termina je ovde, u nedostatku srpske literature, **originalno** skovan).

Literatura je data u redosledu referisanja, i grupisana je (koliko je to bilo moguće) po poglavljima. Na kraju su navedeni neki od važnih linkova "opšte namene", korišćenih praktično u celom radu (ili bar njegovom velikom delu) (odeljak *A Literature*).

1. deo: Digitalizacija i kompresija slika

1. Digitalizacija slika

Digitalizacija je postupak kojim se podaci o kontinualnim pojavama (zvuk, slika, pokret) prevode u digitalni oblik, koji računar kao diskretna mašina jedino može da razume. Srž digitalizacije je *uzorkovanje* – uzimanje uzoraka u diskretnim (vremenskim i/ili prostornim) intervalima. Da bi se digitalizovao *zvuk*, uređaj koji vrši uzorkovanje meri amplitude zvučnih talasa više puta u sekundi. Da bi se digitalizovala *slika*, uzorkuje se boja slike na malim rastojanjima. Da bi se digitalizovao *pokret*, slike se uzorkuju i po vremenskoj komponenti, pa se na ekranu prikazuje više slika u sekundi.

Slika se u računaru predstavlja matricom uzoraka tačaka[1], zvanih *pikseli*. Svaki piksel (*picture element, pixel*) ima svoju boju. Boja se predstavlja u računaru određenim brojem bitova. Za ovakve slike koristi se naziv *bitmapirana (rasterska) grafika*.

O digitalizaciji slika napisano je mnogo tekstova. Neke od knjiga koje se bave ovom problematikom su [1_01, 1_02, 1_03, 1_04]. Tekst iz potpoglavlja 1.1 dat je nešto opširnije u [1_06]. Potpoglavlje 1.2 sadrži **originalni** predlog vizuelizacije matrica, koji je prvi put (zajedno sa tekstom datim u potpoglavljima 3.2.1 i 3.2.2) dat u radu [1_07].

1.1. Optimalni zapis slike

Pri čuvanju slike u računaru stalno se susrećemo sa problemom njenog optimalnog zapisa. Pod optimalnim zapisom smatramo "što verniju sliku, sa što manje zauzetog prostora na disku".

Dva osnovna faktora koji utiču na sadržaj i veličinu grafičkog fajla su *broj piksela slike* i *dubina boje*.

[1] U [1.05] je dato obrazloženje zašto je pravilnije posmatrati piksele kao uzorke tačaka, a ne (uobičajena greška) kao kvadratiće.

1.1.1. Broj piksela slike

Ukupan broj piksela slike je $m \times n$ (m – broj piksela po visini, n – po širini). Koliko piksela ćemo čuvati, zavisi od namene slike.

Ako je jedina planirana namena slike – prikaz na ekranu, onda njena veličina treba da bude takva da se ona u svojoj normalnoj veličini prikaže tako da se cela vidi na ekranu. Slika se u svojoj normalnoj veličini (100%) prikazuje na ekranu tako što se piksel slike predstavi pikselom ekrana.

Rezolucija ekrana se obično definiše kao broj piksela koji se na njemu prikazuje. Tako, ako se matrica piksela na ekranu sastoji iz 768 redova sa po 1024 piksela u redu, kažemo da je rezolucija ekrana 1024×768.

Ponekad se rezolucija ekrana definiše drugačije, kao broj piksela (ili tačaka, dots) po dužnom inču (dpi). Rezolucija većine novih monitora je oko 96 dpi, dok stariji Mac OS monitori imaju rezoluciju od 72 dpi.

Rezolucija štampača je mera broja grafičkih tačaka (dots) po dužnom inču (dpi). Većina stonih laserskih štampača ima rezoluciju od 600 dpi, a grafički štampači (imagesetters) – od 1200 dpi ili više.

Ako želimo da sliku štampamo, treba da čuvamo veći broj piksela slike nego ako samo želimo da je prikažemo na ekranu.

1.1.2. Dubina boje

Informacija o boji svakog piksela slike čuva se u nizu bitova fiksne dužine. Broj bitova upotrebljenih za jedan piksel naziva se *dubina boje* (dubina piksela, bit rezolucija, bit dubina).

Što je dubina boje veća, na slici je moguće prikazati više različitih boja. Odluku o dubini boje, a time i o bogatstvu boja grafike, donosimo u zavisnosti od toga kakvi se podaci na slici nalaze (vodeći, naravno. računa o tome da veća dubina boje znači veći utrošak memorijskog prostora za čuvanje slike).

Ako slika sadrži samo crno–bele elemente (na primer, ako predstavlja skenirani dokument sa tekstom), za opis piksela na njoj dovoljne su dve boje – crna i bela. Ove dve boje mogu se definisati korišćenjem samo jednog bita po pikselu. Zapis grafike sa dubinom boje 1 nazivamo *monohromatska (crno–bela) grafika*. Koristi se i naziv *binarne slike*.

Ako je slikom predstavljen jednostavan crtež, verovatno će dobar izbor biti dubina boje 8 (čime je omogućeno predstavljanje do $2^8 = 256$ različitih boja). Ovakva grafika se obično zadaje *indeksnim slikama*. Vrednosti piksela indeksne slike su zadate indeksima u jednu RGB (R=red, G=green, B=blue) tabelu boja. Ova tabela boja se obično naziva *paleta* ili *CLUT* (Color LookUp Table). Svaki ulaz u CLUT sadrži 24–bitnu vrednost za jednu boju (po 8 bitova za crvenu, zelenu i plavu komponentu). Piksel u indeksnoj kolor slici ukazuje na poziciju unutar CLUT sa podacima o njegovoj boji. Kako je za indeks pozicije u paleti potrebno 8 bitova, broj različitih boja takve slike ograničen je na $2^8 = 256$.

Slike u nijansama sive (grazscale images) dosta se dobro predstavljaju sa 256 nijansi (od crne do bele), što se postiže dubinom boje 8.

Slike u punoj boji mogu se na ekranu prikazati korišćenjem dubine boje 24. U RGB (R=red, G=green, B=blue) modelu boja,[2] to se realizuje tako što se sa po 8 bitova predstavljaju komponente crvene, zelene i plave, koje se kombinuju da bi se prikazala boja piksela. Na ovaj način, na ekranu se može predstaviti 2^{24}, ili približno 16,7 miliona različitih boja, a to je obično više nego dovoljno za ljudsko oko.

Slike koje su pripremljene za štampu u punoj boji mogu se predstaviti u CMYK modelu (C=cyan, M=magenta, Y=yellow, K=key (black)), gde se za svaku od četiri komponente koristi po 8 bitova informacije po pikselu. Dubina boje u ovom slučaju je 32, što bi trebalo da omogući čak 2^{32}, ili preko 4 milijarde različitih boja! U praksi se, međutim, pri štampi ne dobija ni približno toliko boja.

U RGB modelu (aditivan model boja), koji se koristi u prikazu slika na ekranu, koriste se crvena, zelena i plava (boje svetlosti – što se više boje dodaje, to je ukupna boja svetlija). S druge strane, tehnika štampe se zasniva na CMYK modelu (subtraktivan model), gde se boje koriste kao pigmenti; što se više boje stavi, dobija se tamnija boja. Tehnika štampe je prljava tehnika, pa je broj boja koje se mogu odštampati čak i manji od broja boja koje se RGB modelom mogu prikazati na ekranu.

---

[2] *Model boja* je sistem za stvaranje celog raspona boja iz malog skupa primarnih boja. Postoje dva tipa modela boja, subtraktivni i aditivni. *Aditivni* modeli boja koriste svetlost da bi prikazali boju, a *subtraktivni* modeli koriste štamparska mastila. Boje koje se vide u aditivnim modelima rezultat su propuštene svetlosti, dok su boje u subtraktivnim modelima – rezultat odbijene svetlosti.

1.1.3. Grafički fajl formati

Format grafičkog fajla određuje način na koji će informacija o slici biti organizovana. Danas postoji vrlo veliki broj različitih grafičkih formata. Međutim, bez obzira na to koji format fajla se koristi, memorijski prostor potreban za smeštanje podataka o slici približno se računa množenjem broja piksela slike sa dubinom boje.

Neki od ovih formata su namenjeni editovanju slika, neki za njihovo arhiviranje i prikaz na Web–u. Neki su dobri za jednostavnu grafiku, neki za fotografije. Spomenimo neke, najčešće korišćene.

PCX, TIFF i BMP formati široko su zastupljeni u obradi slika, uključujući skeniranje, prenos među platformama i njihovo korišćenje u stonom izdavaštvu. Sva ova tri formata sadrže podatke koji su ili nekomprimovani, ili se komprimuju bez gubitaka, što ih čini dobrim pri editovanju, ali ih diskvalifikuje za korišćenje na Web–u.

Formati GIF, JPEG i PNG su pre svega namenjeni korišćenju na Web–u, jer zahvaljujući moćnim tehnikama kompresije koji se u njima koriste, troše manje prostora za podatke o slikama, pa se lakše šalju preko mreže. Ova tri formata imaju još jednu važnu osobinu koju prva tri nemaju, koja ih dodatno kvalifikuje za Web, a to je mogućnost progresivnog prikaza.[3]

1.2. Slika i njena matrica. Matrica i njena slika

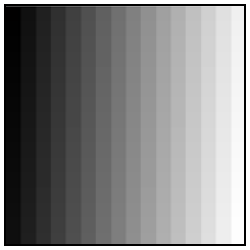
Digitalizacijom se slici u nijansama sive dodeljuje matrica. Dimenzija te matrice određena je brojem piksela slike. Elementi matrice slike su brojevi iz skupa $\{0,1,2,...,255\}$. Element ima vrednost 0, ako je odgovarajući piksel crn; vrednost 255, ako je beo. Brojevi između 0 i 255 odgovaraju nijansama sive – što je broj veći, piksel je svetliji.

Slike u punoj boji (RGB model boja) u računaru se predstavljaju sa tri slike u nijansama sive (po jedna za svaku – crvenu, zelenu i plavu – komponentu boje).

Postoji obostrano jednoznačno preslikavanje skupa slika u nijansama sive, u skup matrica čiji elementi uzimaju celobrojne vrednosti od 0 do 255.

[3] Termin *progresivni prikaz* odnosi se na mogućnost da se slika na mreži, umesto red po red (uobičajeni način) prikazuje kao celina, ali prvo u obliku slike slabije rezolucije, da bi zatim u jednom ili nekoliko prolaza bila popravljana, i na kraju se prikazala kao slika u punoj rezoluciji.

Sledi prikaz (zumirane i u obliku matrice) slike dimenzije 16×16, sa svim mogućim nijansama sive boje [4]



0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

Svaka operacija obrade slike u računaru može se posmatrati kao operacija nad njenom matricom. Rezultat je obično matrica dimenzije originalne slike ($m \times n$), čiji elementi ne moraju biti iz skupa $\{0,1,...,255\}$, nego to mogu biti proizvoljni realni, pa i kompleksni brojevi.

Tako, matrica dobijena *diskretnom kosinusnom transformacijom* matrice slike (jedan od koraka u JPEG kompresiji, detaljnije opisan kasnije), ima ravnomerno zastupljene pozitivne i negativne vrednosti, sa vrlo velikim rasponima u magnitudi.

Matrica koja se dobija primenom *Furijeove transformacije* je kompleksna, sa opet vrlo velikim rasponima u magnitudi.

Matrica *belog Gausovog šuma* ima ravnomerno zastupljene pozitivne i negativne (realne) elemente, sa malim magnitudama.

Matrice koje se ovde koriste su vrlo visokih dimenzija (nekoliko desetina ili stotina hiljada, pa čak i miliona elemenata). Matricu ovolikih dimenzija čoveku je nemoguće da prati. Da bi se stekao uvid u njen sadržaj, prirodno se nameće potreba da se ona vizuelizuje.

[4] U slikama koje ovaj tekst prate, ukoliko je potrebno vizuelno razdvojiti sliku od pozadine, postoji bordura oko slike. Takođe, velike matrice uz sliku obično su predstavljene bez odgovarajućih uglastih zagrada.

Ova ideja, svakako, nije nova. U programima namenjenim za rad sa matricama, postoji mogućnost njihovog predstavljanja slikom. Međutim, ova prezentacija je uglavnom neadekvatna, jer ne uzima u obzir vrednosti izvan skupa $\{0,1,2,...,255\}$.

Matlab i slični programi predstavljaju proizvoljne matrice na sledeći način:

- Prvo, odbacuje se imaginarni deo (ako je matrica kompleksna)
- Zatim, svi elementi se zaokružuju na najbliži ceo broj
- Najzad, vrednosti manje od 0 postaju 0; vrednosti veće od 255 postaju 255

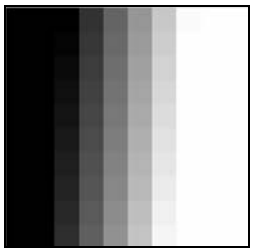
Na primer, matrica

```
-100 -50   0  50  100  150  200  250  300  350
 -95 -45   5  55  105  155  205  255  305  355
 -90 -40  10  60  110  160  210  260  310  360
 -85 -35  15  65  115  165  215  265  315  365
 -80 -30  20  70  120  170  220  270  320  370
 -75 -25  25  75  125  175  225  275  325  375
 -70 -20  30  80  130  180  230  280  330  380
 -65 -15  35  85  135  185  235  285  335  385
 -60 -10  40  90  140  190  240  290  340  390
 -55  -5  45  95  145  195  245  295  345  395
```

da bi se mogla predstaviti slikom, menja se u

```
  0   0   0  50  100  150  200  250  255  255
  0   0   5  55  105  155  205  255  255  255
  0   0  10  60  110  160  210  255  255  255
  0   0  15  65  115  165  215  255  255  255
  0   0  20  70  120  170  220  255  255  255
  0   0  25  75  125  175  225  255  255  255
  0   0  30  80  130  180  230  255  255  255
  0   0  35  85  135  185  235  255  255  255
  0   0  40  90  140  190  240  255  255  255
  0   0  45  95  145  195  245  255  255  255
```

i prikazuje kao:



Jasno, takav prikaz ne pokazuje dobro informaciju o sadržaju matrice. Dobar prikaz matrice treba da omogući predstavljanje prirode i rasporeda elemenata u njoj: on treba da pokaže da li su elementi matrice pozitivni ili negativni, realni ili kompleksni, kao i kakav je odnos u magnitudama elemenata.

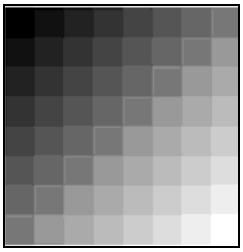
Tekst koji sledi daje (**originalan**) predlog takve vizuelizacije proizvoljnih matrica.

1.2.1. Prikaz realne matrice

Kod matrice čiji su svi elementi ≥ 0, vrednost 0 možemo predstaviti crnom, a maksimalnu pozitivnu vrednost – belom bojom; tako se pozitivne vrednosti matrice prikazuju kao nijanse sive (od crne za nulu, do bele za maksimalnu pozitivnu vrednost).

Na primer – matrica i slika:

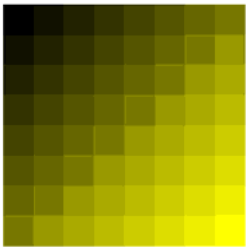
0	50	100	150	200	250	300	350
50	100	150	200	250	300	350	400
100	150	200	250	300	350	400	450
150	200	250	300	350	400	450	500
200	250	300	350	400	450	500	550
250	300	350	400	450	500	550	600
300	350	400	450	500	550	600	650
350	400	450	500	550	600	650	700



U ovom rešenju prvo se skaliraju vrednosti matrice da bi se dovele u razmak [0,255] pre prikaza – sve se vrednosti množe faktorom (255/maks.element) (u ovom primeru, to je 255/700).

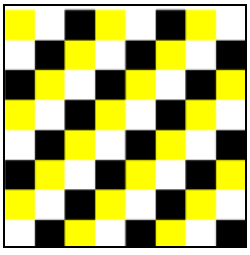
Matrica sa svim elementima manjim ili jednakim 0 može biti prikazana slično, u nijansama od crne do žute:

0	-50	-100	-150	-200	-250	-300	-350
-50	-100	-150	-200	-250	-300	-350	-400
-100	-150	-200	-250	-300	-350	-400	-450
-150	-200	-250	-300	-350	-400	-450	-500
-200	-250	-300	-350	-400	-450	-500	-550
-250	-300	-350	-400	-450	-500	-550	-600
-300	-350	-400	-450	-500	-550	-600	-650
-350	-400	-450	-500	-550	-600	-650	-700



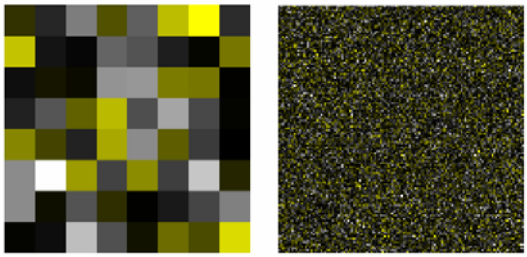
Matrica koja ima i pozitivne i negativne vrednosti prikazaće se slikom, u kojoj će pozitivne vrednosti biti u nijansama od crne do bele, a negativne – od crne do žute. Ako je maksimalna apsolutna vrednost pozitivnih elemenata veća od one za negativne elemente, na slici će biti neki piksel beo (a ni jedan sasvim žut) (važi i obrnuto). Ako su maksimalne apsolutne vrednosti jednake za pozitivne i negativne brojeve, postojaće i sasvim beli, i sasvim žuti pikseli:

-1	1	0	-1	1	0	-1	1
1	0	-1	1	0	-1	1	0
0	-1	1	0	-1	1	0	-1
-1	1	0	-1	1	0	-1	1
1	0	-1	1	0	-1	1	0
0	-1	1	0	-1	1	0	-1
-1	1	0	-1	1	0	-1	1
1	0	-1	1	0	-1	1	0



Na ovaj način moguće je predstaviti i matrice čiji su elementi dosta veći od 255, a da se i dalje uočava koji elementi su veći, a koji manji. Takođe, i matrice sa vrlo malim elementima ovde se mogu prikazati tako da slika ne bude crna.

Na sledećoj slici predstavljene su dve matrice belog Gausovog šuma. Matrica levo je znatno manjih dimenzija u odnosu na onu s desne strane, pa su zato njeni elementi predstavljeni "zumirano".



1.2.2. Prikaz kompleksne matrice

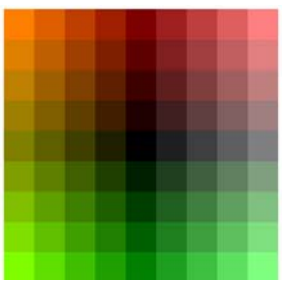
Čisto imaginarni brojevi mogu se predstaviti na sličan način. Negativne imaginarne vrednosti prikazujemo zelenom, a pozitivne – crvenom bojom.

Na sledeće tri slike predstavljene su realna i imaginarna matrica, i njihova suma – kompleksna matrica.

```
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
-4  -3  -2  -1  0  1  2  3  4
```

```
+4i +4i +4i +4i +4i +4i +4i +4i  +4i
+3i +3i +3i +3i +3i +3i +3i +3i  +3i
+2i +2i +2i +2i +2i +2i +2i +2i  +2i
+1i +1i +1i +1i +1i +1i +1i +1i  +1i
 0   0   0   0   0   0   0   0    0
-1i -1i -1i -1i -1i -1i -1i -1i  -1i
-2i -2i -2i -2i -2i -2i -2i -2i  -2i
-3i -3i -3i -3i -3i -3i -3i -3i  -3i
-4i -4i -4i -4i -4i -4i -4i -4i  -4i
```

```
-4+4i -3+4i -2+4i -1+4i +4i 1+4i 2+4i 3+4i 4+4i
-4+3i -3+3i -2+3i -1+3i +3i 1+3i 2+3i 3+3i 4+3i
-4+2i -3+2i -2+2i -1+2i +2i 1+2i 2+2i 3+2i 4+2i
-4+1i -3+1i -2+1i -1+1i +1i 1+1i 2+1i 3+1i 4+1i
-4    -3    -2    -1     0  1    2    3    4
-4-1i -3-1i -2-1i -1-1i -1i 1-1i 2-1i 3-1i 4-1i
-4-2i -3-2i -2-2i -1-2i -2i 1-2i 2-2i 3-2i 4-2i
-4-3i -3-3i -2-3i -1-3i -3i 1-3i 2-3i 3-3i 4-3i
-4-4i -3-4i -2-4i -1-4i -4i 1-4i 2-4i 3-4i 4-4i
```



Svakako, predložena kombinacija boja nije jedina moguća. Međutim, ona nije ni potpuno proizvoljna. U izboru boja vrlo je važno da dve udaljene tačke u kompleksnoj ravni budu predstavljene različitim bojama. Ustvari, **između bilo koje dve tačke u kompleksnoj ravni predstavljene istom bojom ne sme postojati tačka obojena drugom bojom**. (Jasno, dve bliske tačke, zbog beskonačnog broja tačaka u kompleksnoj ravni i konačnog broja različitih boja, mogu biti iste boje).

Predložena kombinacija boja (žuto–belo & zeleno–crveno) je korektna u tom smislu. Intenzitet korišćenih boja raste udaljavanjem od koordinatnog početka, i zato, ni u jednom kvadrantu kompleksne ravni ne postoje dve udaljene tačke obojene istom bojom.

Uz to, tačke (x, y) su predstavljene (po kvadrantima) u sledećim bojama (žuto=*(r+g)*, belo=*(r+g+b)*):

- I kvadrant: $|x| \cdot (r+g+b) + |y| \cdot r$ najviše crvene, i nešto plave boje
- II kvadrant: $|x| \cdot (r+g) + |y| \cdot r$ najviše crvene, i bez plave boje
- III kvadrant: $|x| \cdot (r+g) + |y| \cdot g$ najviše zelene, i bez plave boje
- IV kvadrant: $|x| \cdot (r+g+b) + |y| \cdot g$ najviše zelene, i nešto plave boje

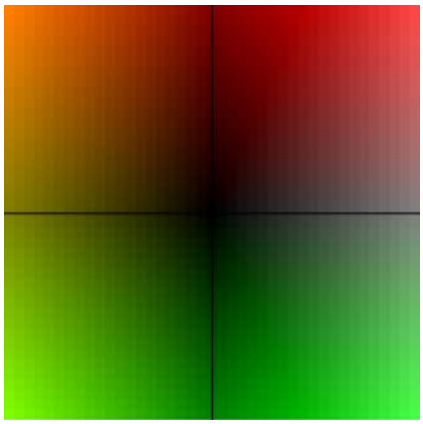
Zato, ne postoje ni dve tačke u različitim kvadrantima obojene istom bojom.

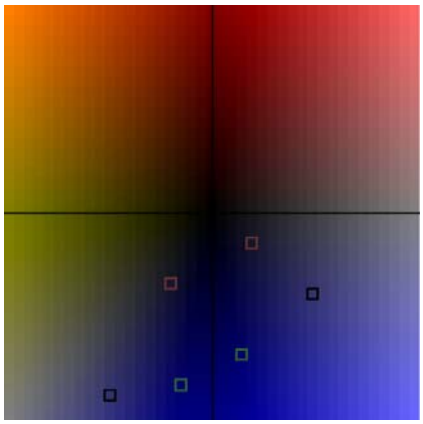
Da smo umesto zelene koristili plavu boju, postojale bi dve tačke $(-x,-y)$ i $(x, x-y)$ $(x, y > 0, y > x)$ (kombinacija žuto–belo & plavo–crveno):

- III kvadrant: $x \cdot (r+g) + y \cdot b = x \cdot (r+g+b) + (y-x) \cdot b$
- IV kvadrant: $x \cdot (r+g+b) + (y-x) \cdot b$

Tako, u trećem i četvrtom kvadrantu postoje dve tačke predstavljene istom bojom (kombinacijom bele i plave). Zato, kombinacija žuto–belo & plavo–crveno nije dobra.

Na sledeće dve slike prikazane su navedene dve kombinacije boja. Na slici levo je "dobra" kombinacija (žuto–belo & zeleno–crveno). Desno je "loša" kombinacija (žuto–belo & plavo–crveno). Tu je obeleženo nekoliko parova udaljenih tačaka, obojenih istom bojom.





U daljem tekstu će se osim Matlab-ove funkcije *imshow*, za prikaz slika–matrica koristiti funkcija *boje*:

```
% boje: funkcija za prikaz slike matrice (realne ili kompleksne)
function boje(mat,mp,np,ind)
% prikaz kompleksne slike
[m n]=size(mat);
matr=real(mat);
mati=imag(mat);
smatr=zeros(m,n,3);smrp=smatr;smrm=smatr;
smati=zeros(m,n,3);smip=smati;smim=smati;
for i1=1:m
    for i2=1:n
        if matr(i1,i2)>0
            smrp(i1,i2,1)=matr(i1,i2);
            smrp(i1,i2,2)=matr(i1,i2);
            smrp(i1,i2,3)=matr(i1,i2);
        else
            smrm(i1,i2,1)=-matr(i1,i2);
            smrm(i1,i2,2)=-matr(i1,i2);
        end
        if mati(i1,i2)>0
            smip(i1,i2,1)=mati(i1,i2);
        else
            smim(i1,i2,2)=-mati(i1,i2);
        end
    end
end
smr=smrp+smrm;
smi=smip+smim;
sm=smr+smi;
ma=max(max(max(sm)));
%slika mora ima vrednosti piksela izmedju 0 i 1
%if ma>1
    sm=sm/ma;
    %end
subplot(mp,np,ind);imshow(sm)
```

2. Kompresija podataka

Veličina memorijskog prostora potrebnog za zapis bitmapirane slike približno iznosi

$$broj\ piksela\ slike \cdot dubina\ boje$$

To je često neprihvatljiva količina prostora. Neprihvatljiva za čuvanje na disku, a još neprihvatljivija ako fajl treba slati preko mreže. U pokušaju da se reše problemi koje stvaraju veliki grafički (odnosno muzički, video,...) fajlovi, pojavile su se različite tehnike kompresije podataka.

Pod kompresijom podrazumevamo smeštanje podataka na način koji će smanjiti potrošnju prostora.

U ovom poglavlju, prvo se uvode pojmovi: redundanca, komunikacija, količina informacije, entropija. U nastavku poglavlja opisuju se neki važni algoritmi kompresije bez gubitaka.

U izučavanju teme ovog poglavlja mogle bi biti korisne knjige [2_01, 2_02].

2.1. Redundanca i kompresija

Svaki signal realnog sveta ima dve osnovne komponente:

- bitni deo signala – nepredvidljiv, nepoznat deo; ova komponenta, koja predstavlja pravu informaciju u signalu, naziva se *entropija*;
- nebitni deo signala je onaj deo koji se može predvideti iz entropije – ova komponenta naziva se *redundanca*.

Postoje dva tipa redundance – *statistička* i *psihovizuelna*.

U statističku redundancu spadaju *međupikselska* i redundanca *kodiranja*. Međupikselska redundanca postoji zato što pikseli na slici, a takođe i pikseli u grupi uzastopnih slika ili video kadrova, nisu statistički nezavisni. Ona se može podeliti u dve kategorije, na *prostornu* i *vremensku* redundancu. Zbog međupikselske redundance, nije neophodno da se predstavi svaki poseban piksel slike. Vrednost piksela moguće je proceniti na osnovu vrednosti susednih piksela.

Redundanca kodiranja nije vezana za redundancu same informacije, nego za prikaz informacije, tj. samo kodiranje. Često se može postići ušteda u prostoru, ako se umesto koda fiksne dužine, upotrebi neki od tzv. *entropijskih* kodova, promenljive dužine kodnih reči.

Psihovizuelna redundanca tiče se informacije na koju ljudska čula nisu osetljiva. Takvu informaciju moguće je neprimetno ukloniti iz fajla i tako smanjiti količinu podataka potrebnu za njegov zapis.

Kompresijom se iz signala uklanja redundanca. Podaci se komprimuju pri smeštanju na disk. Komprimovani se šalju i preko mreže. Da bi se upotrebili ponovo (bilo na istom, bilo na drugom računaru), moraju se dekomprimovati. Pri dekompresiji se redundantna informacija ponovu ubacuje u signal. **Čoveku je redundanca u podacima potrebna, da bi ih mogao razumeti**.

Sve tehnike kompresije mogu se podeliti u dve grupe: tehnike kompresije *bez gubitaka* i one *sa gubicima*.

Kompresionim tehnikama bez gubitaka veličina fajla smanjuje se tako da se kasnije, po dekomprimovanju, dobija fajl koji je identičan originalu. Drugim rečima, veličina fajla se smanjuje bez žrtvovanja ijednog prvobitnog podatka. **Tehnike kompresije bez gubitaka uklanjaju** iz podataka **statističku redundancu**.

Metode kompresije bez gubitaka moguće je podeliti u tri grupe:

- Entropijsko kodiranje,
- Kodiranje dužine sekvence (run length encoding, RLE)
- Adaptivni rečnički algoritmi, kao što je LZW

Kompresija sa gubicima pravi daleko manje fajlove nego što se mogu dobiti nekom kompresionom tehnikom bez gubitaka. Pri tome se dobija fajl koji nije identičan polaznom fajlu, ali su gubici uglavnom neprimetni za ljudsko oko (ili, u slučaju zvučnih fajlova, za ljudsko uho). Ove tehnike baziraju se na **uklanjanju psihovizuelne redundance** iz signala.

Kompresiju sa gubicima moguće je primeniti samo na neke tipove podataka – na grafiku, audio i video. Za podatke i programe možemo koristiti samo kompresiju bez gubitaka.

2.2. Informacija, komunikacija, entropija

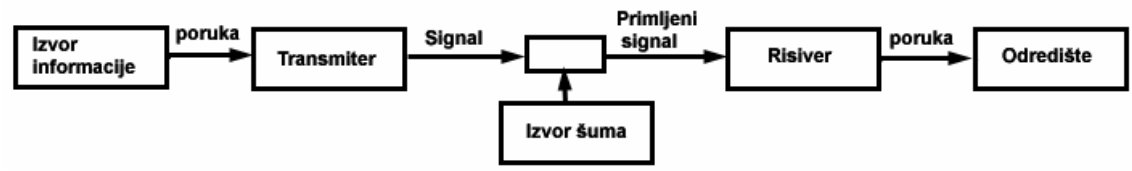
2.2.1. Šenonova teorija informacije

Teorija informacije (naziva se i *teorija komunikacije*) je grana matematike koju je zasnovao **Šenon** (Claude Shannon) u kasnim četrdesetim godinama prošlog veka [2_03, 2_04].

Šenon *komunikaciju* definiše kao **razmenu informacija između dve tačke**. Informacija se šalje od izvora, ka odredištu. Na tom putu ona prolazi kroz komunikacioni kanal. Ukoliko u komunikacionom kanalu postoji šum, informacija koja stigne na odredište može da ima izvesne greške.

Tako, osnovni problem komunikacije je poruku odabranu u jednoj tački reprodukovati, tačno ili približno, u drugoj tački.

Takva komunikacija se može predstaviti pomoću *komunikacionog sistema* (slika 2.1).



Slika 2.1: Shematski dijagram opšteg komunikacionog sistema

Ovaj sistem ima pet delova:

1. *Izvor informacije* proizvodi poruku koju treba preneti do odredišta.
2. *Transmiter* poruku prevodi u signal pogodan za prenos preko kanala.
3. *Kanal* je medijum koji se koristi da bi se signal preneo od transmitera do risivera.
4. *Risiver* obavlja inverznu operaciju od one koju je obavio transmiter, rekonstruišući poruku iz signala.
5. *Odredište* je primalac poruke.

Ovakav sistem može se pronaći u mnogim situacijama. U svakodnevnom životu, oblici komunikacije su: govor/slušanje; čitanje/pisanje; pokreti tela; pokazivanje naših osećanja... S druge strane, i kompresiju – dekompresiju, a i praksu žiga, možemo shvatiti kao komunikaciju.

U svetlosti kompresije – dekompresije podataka kao digitalnog komunikacionog sistema, mi podatke fajla tretiramo kao digitalne poruke koje treba kodirati, tako da se utroši što manje prostora.

Kodiranje poruka je proces u kome se svakoj digitalnoj poruci dodeljuje odgovarajući kôd. U slučaju kodiranja u računaru, u kome osnovu predstavlja binarni brojni sistem, poruka će biti kodirana sekvencom binarnih cifara, tj. nula i jedinica.

Za optimalno kodiranje, važno je opisati količinu informacije sadržane u pojedinim porukama. Jasno, od interesa je da kodiranje bude takvo da se različitim porukama dodele različiti kodovi (da bi kasnije dekodiranje bilo moguće) i da ukupna očekivana dužina sekvence u kodu poruke bude što je moguće manja. Tako se nailazi na problem merenja količine informacije u porukama. Ovde treba voditi računa da budu zadovoljena dva zahteva:

1. Mera za količinu informacije treba da monotono opada s porastom verovatnoće poruke. Drugim rečima, što je njena verovatnoća veća, to poruka sadrži manju količinu informacije.
2. Ako neko pošalje dve poruke, onda ukupna količina informacije treba da bude jednaka zbiru količina informacije svake od poruka. Jasno, ako su dve poruke statistički nezavisne, onda je verovatnoća kombinacije jednaka proizvodu verovatnoća pojedinih poruka.

Ako informacioni sadržaj (količinu informacije) poruke x definišemo izrazom:

$$I_x = \log_a \frac{1}{p_x} = -\log_a p_x \qquad (2.1)$$

gde je p_x verovatnoća poruke x, a $a > 1$, jasno je da će navedeni zahtevi biti zadovoljeni.

Jedinica količine informacija naziva se

- *Shannon* (ili *bit*) ukoliko je korišćena osnova logaritma $a = 2$,
- *Hartley* (ili *digit*) za slučaj osnove logaritma $a = 10$ i
- *nit* za slučaj prirodnog logaritma (osnova e).

Uobičajeno je da se koristi logaritam sa osnovom 2, jer na taj način količina informacije sadržana u poruci predstavlja broj bitova potrebnih za opis poruke u računaru.

Ako imamo k mogućih simbola, $(s_i, i = 1,...,k)$, od kojih svaki ima svoju verovatnoću pojavljivanja p_i ($\sum_{i=1}^{k} p_i = 1$), informacioni sadržaj svakog od simbola s_i definisan je sa

$$I_i = \log_2 \frac{1}{p_i} = -\log_2 p_i . \qquad (i = 1,2,...,k) \qquad (2.2)$$

Primer 2.1: Pri bacanju novčića moguća su (i jednako verovatna) dva ishoda: "pismo" i "glava". Ova dva ishoda mogu se redom kodirati bitovima 0 i 1. Količina informacije poruke o svakom od ova dva ishoda je

$$-\log_2\left(\frac{1}{2}\right) = \log_2 2 = 1.$$

Ako se istovremeno bacaju tri novčića, moguće ishode (kombinaciju tri statistički nezavisna događaja) moguće je kodirati trojkama bitova: 000, 001, 010, 011, 100, 101, 110, 111. Količina informacije poruke u svakom od ovih ishoda je

$$3\cdot\left(-\log_2\left(\frac{1}{2}\right)\right) = 3\log_2 2 = 3.$$

Primer 2.2: Ako uređaj može da proizvede tri različita simbola, A, B, C, sa jednakim verovatnoćama, 1/3, informacioni sadržaj za svaki od ta tri ishoda jednak je $\log_2 3$. Ako se ovaj uređaj kombinuje sa uređajem koji proizvodi dva različita događaja, 1 i 2, sa jednakim verovatnoćama, 1/2 (informacioni sadržaj svakog od mogućih ishoda ovog drugog uređaja je $\log_2 2 = 1$), onda je informacioni sadržaj svakog mogućeg ishoda složenog događaja – istovremenog izvršavanja ova dva uređaja (mogući ishodi su A1, A2, B1, B2, C1, C2, gde svaki od ovih 6 ishoda ima verovatnoću 1/6) jednak $\log_2 3 + \log_2 2 = \log_2(3\cdot 2) = \log_2 6$.

Što je verovatnoća pojavljivanja simbola s_i manja, to je njegov informacioni sadržaj veći. Informacioni sadržaj sigurnog događaja (odnosno simbola koji se pojavljuje sa verovatnoćom 1) je $= 0$. S druge strane, ako je verovatnoća događaja (pojave simbola) mala, informacioni sadržaj takvog događaja je velik.

2.2.2. Entropija izvora

U komunikacionim sistemima obično imamo prenošenje **dugih sekvenci** simbola od izvora informacija do odredišta. Zato je mnogo važnije analizirati usrednjenu informaciju koju izvor generiše, nego pratiti informacioni sadržaj jednog simbola.

Srednja vrednost količine informacija $I(x_i)$ u okviru azbuke od k različitih simbola $x_1,...,x_k$ sa verovatnoćama redom $p_1,...,p_k$, koja karakteriše određeni izvor X može se zapisati na sledeći način:

$$H(X) = E[I(x_i)] = \sum_{i=1}^{k} p_i I(x_i) = -\sum_{i=1}^{k} p_i \log_2 p_i \qquad (2.3)$$

Vrednost $H(X)$ naziva se *entropija izvora* X. Predstavlja **meru usrednjenog informacionog sadržaja** koji generiše izvor **po simbolu**.

Da bismo izbegli slučajeve nedefinisane vrednosti H ako je za neko i, $p_i = 0$, možemo definisati $0 \log_2 0 = 0$.

$H = 0$ ako i samo ako su svi p_i sem jednog jednaki nuli, a taj jedan ima vrednost 1. U svim drugim slučajevima, H je pozitivno.

Teorema: Entropija je najveća ako svi simboli imaju jednake verovatnoće.

Dokaz: Ako k različitih simbola imaju verovatnoće pojavljivanja redom

$$p_1, p_2, ..., p_k \ (p_i \in (0,1], \sum_{i=1}^{k} p_i = 1),$$

entropija će biti

$$H = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i$$

Ako su verovatnoće $p_3, ..., p_k$ zadate, kakve treba da budu verovatnoće p_1 i p_2 da bi entropija bila maksimalna?

$$p_1 = x, \ p_2 = 1 - x - p_3 - ... - p_k$$

$$H = -x \cdot \log_2 x - (1 - x - p_3 - ... - p_k) \cdot \log_2 (1 - x - p_3 - ... - p_k) - \sum_{i=3}^{k} p_i \cdot \log_2 p_i$$

$$H' = -\log_2 x - x \cdot (1/x) \cdot \ln 2 + \log_2 (1 - x - p_3 - ... - p_k) +$$
$$+ (1 - x - p_3 - ... - p_k) \cdot (1/(1 - x - p_3 - ... - p_k)) \cdot \ln 2 - 0 = -\log_2 x + \log_2 (1 - x - p_3 - ... - p_k)$$

$H' = 0$ za $x = 1 - x - p_3 - ... - p_k$, tj.

$p_1 = x = (1 - p_3 - ... - p_k)/2$,

$p_2 = 1 - p_3 - ... - p_k - (1 - p_3 - ... - p_k)/2 = (1 - p_3 - ... - p_k)/2$

$H' = 0$ za $p_1 = p_2$

$H'' = -1/x \cdot \ln 2 - 1/(1 - x - p_3 - ... - p_k) \cdot \ln 2 = -\ln 2 \cdot (1/x + 1/(1 - x - p_3 - ... - p_k)) \le 0 \Rightarrow$

$\Rightarrow$ Entropija dostiže maksimum za $p_1 = p_2$

Isti zaključak imamo i ako smo umesto za p_1 i p_2, uz fiksirane ostale verovatnoće tražili odnos bilo koje dve verovatnoće p_j i p_l ($(j, l \in \{1,2,...,k\})$ – entropija ce biti najveća ako je $p_j = p_l$;

Tako, entropija dostiže maksimum za $p_1 = p_2 = ... = p_k = p = 1/k$. Formula za entropiju u ovom slučaju je

$H = -\log_2 p = \log_2 k$

2.2.3. Entropija slike

Entropija slike je mera "dešavanja" na slici. Sasvim ravna (jednobojna) slika ima entropiju 0. Slike sa malom entropijom mogu se komprimovati (bez gubitaka) na relativno malu veličinu.

Neka je slika S data matricom piksela dimenzije $m \times n$. Neka se na njoj pojavljuje k različitih boja, $s_1,...,s_k$, redom $d_1,...,d_k$ puta, i važi $\sum_{i=1}^{k} d_i = m \cdot n$.

Verovatnoća pojavljivanja p_i boje s_i $(i = 1,...,k)$ na slici može se definisati sa

$$p_i = \frac{d_i}{m \cdot n} \tag{2.4}$$

Tada za sliku važi definicija entropije

$$H(S) = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i \tag{2.5}$$

Entropija slike se izražava **brojem bitova po pikselu**.

Primer 2.3: Slika dimenzije $m \times n$, koja je cela obojena jednom bojom, ima za tu (jedinu) boju verovatnoću $p_1 = \frac{d_1}{mn} = \frac{mn}{mn} = 1$. Entropija ove slike je

$H = -p_1 \log_2 p_1 = -1 \log_2 1 = 0$

Primer 2.4: Entropija crno–bele slike, sa $p_1 = p_2 = \frac{1}{2}$ (isti broj crnih i belih piksela) je:

$H = -p_1 \log_2 p_1 - p_2 \log_2 p_2 = \frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2 = 1$

Primer 2.5: Entropija crno–bele slike, sa 75% belih, i 25% crnih piksela ($p_1 = \frac{3}{4}, p_2 = \frac{1}{4}$) je:

$H = -\frac{3}{4}\log_2 \frac{3}{4} - \frac{1}{4}\log_2 \frac{1}{4} = -\frac{3}{4}(\log_2 3 - 2) - \frac{1}{4}(-2) = 2 - \frac{3}{4}\log_2 3 = 0.8113$

Primer 2.6: Slika u nijansama sive, na kojoj se svih 256 boja pojavljuje jednak broj puta ($p_1 = ... = p_{256} = \frac{1}{256}$) ima entropiju

$$H = 256 \cdot \frac{1}{256} \cdot \log_2 256 = 8 \text{ bitova po pikselu}$$

Na entropiju slike ne utiče njena veličina, ni raspored boja. **Na entropiju slike utiču samo broj različitih boja na slici i njihove frekvencije ("verovatnoće")**. Iz prethodnih primera se vidi da entropija direktno odražava sadržaj informacija na slici. **Entropija slike je maksimalna kada svaka boja na njoj ima istu frekvenciju.**

Šenon je dokazao da **entropija predstavlja granicu kompresije** podataka bez gubitaka.

Njegova teorema kodiranja izvora bez šuma tvrdi da je za diskretan, stacionaran (stabilan) izvor informacija bez memorije, minimalni broj bitova po simbolu potreban za kodiranje simbola, jednak entropiji izvora.[5]

Ova teorema daje **donju granicu u kodiranju izvora**. Šenon je pokazao da se entropija pri kodiranju može dostići kada kašnjenje kodiranja teži beskonačnosti. Pod kašnjenjem kodiranja smatra se da koder čeka i zatim kodira izvestan broj simbola odjednom. Na sreću, i sa konačnim kašnjenjem kodiranja, može se postići srednja dužina kodne reči vrlo bliska entropiji.[6]

To u kontekstu slika znači da u slučaju crno–bele slike sa $p_1 = p_2 = 0.5$ ne postoji način da se ta slika bez gubitaka informacije zapiše u manje od mn bitova[7]. Jedno rešenje zapisa sa mn bitova je da se crni pikseli kodiraju binarnom nulom, a beli – jedinicom.

U slučaju da je verovatnoća crne i bele boje redom, 0.25 i 0.75, moguće je primeniti malopre navedeno kodiranje, i na taj način će se slika dimenzije $m \times n$ kodirati sa mn bitova. Međutim, po Šenonu, postoji kodiranje koje će omogućiti da se bez gubitaka u podacima, slika zapiše sa približno $0.8mn$ bitova. Problem je da se nađe takvo kodiranje.

Sledi primer koji pokazuje raspored vrednosti entropije za blokove veličine 8×8 piksela za sliku 'Cameraman' (slika 2.2). Desno su prikazane vrednosti entropije za

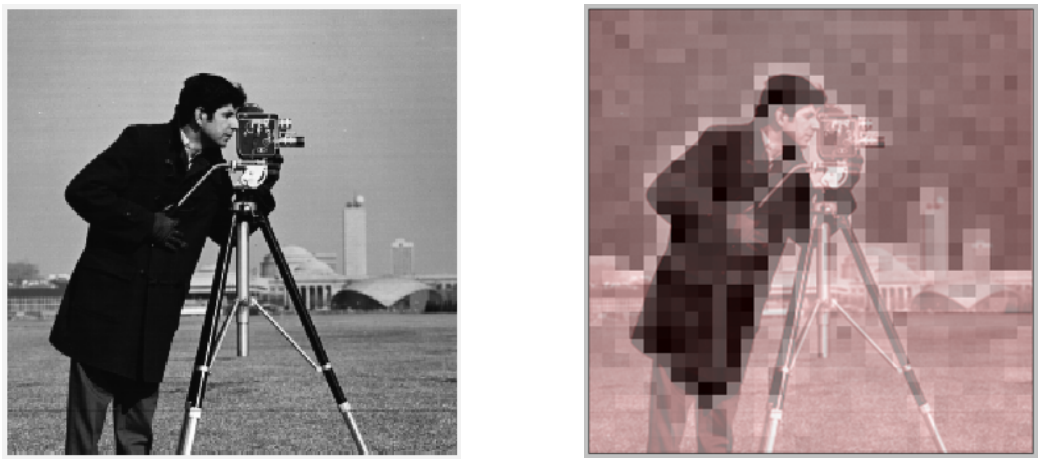
---

[5] Pod "diskretan", podrazumeva se da je izvor prebrojiv skup simbola. "Bez memorije" znači da pojavljivanje simbola u skupu ne zavisi od prethodnog simbola.

[6] Pretpostavke ove teoreme u praksi obično nisu sasvim zadovoljene. Pre svega, slika nije "bez memorije", jer piksel zavisi od okolnih piksela. S druge strane, i pretpostavka stacionarnosti nije zadovoljena u praksi. Tako, Šenonova teorema predstavlja samo teoretski putokaz. Nema, međutim, sumnje da je ona osnovni teoretski rezultat u teoriji informacije.

[7] Ustvari, može da postoji, ako slika nije "bez memorije". Na primer, crno-bela slika dimenzije $m \times n$ u kojoj se crni i beli pikseli naizmenično (pravilno) menjaju, može se predstaviti sa mnogo manje od mn bitova.

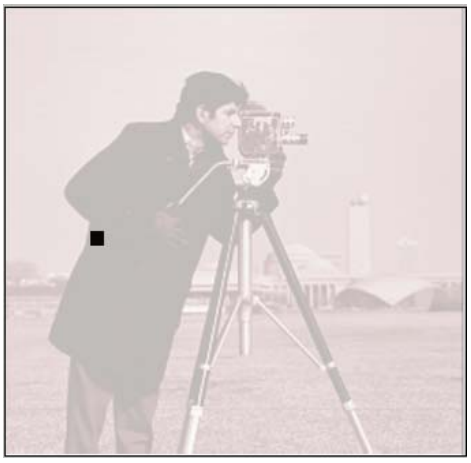
blokove slike. Oblasti niže entropije su označene tamnijom, a oblasti više entropije – svetlijom bojom.



Slika 2.2: Raspored entropije po blokovima slike (dimenzije blokova su 8×8 piksela)

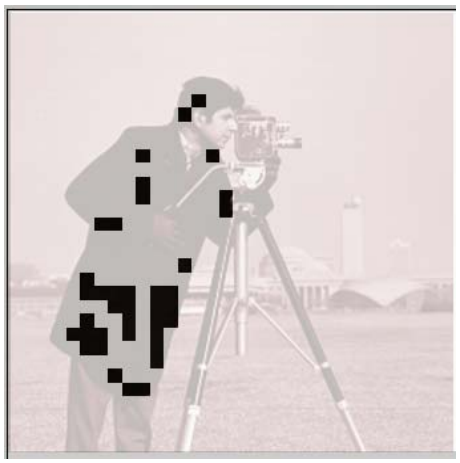
Entropija bloka slike govori o meri koliko se u njemu "nešto dešava". Entropija takođe ima veće vrednosti na konturama, ali i u drugim oblastima koje nisu "jednobojne". Tako je na ovoj slici entropija dosta velika u oblastima gde su prikazani lice, zgrade, trava, fotoaparat,... Mala je kod prikaza neba, i posebno, kod kaputa.

Sledi još jedna ilustracija vrednosti entropije po blokovima (slika 2.3).

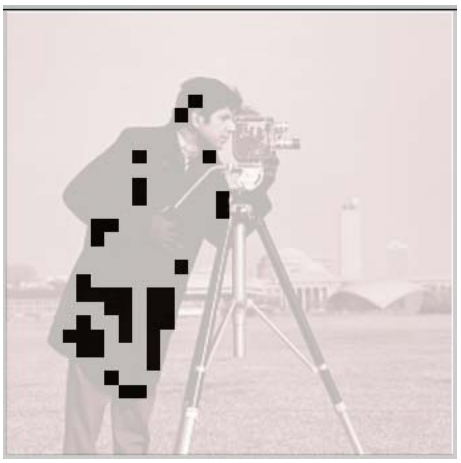


Entropija 0–1

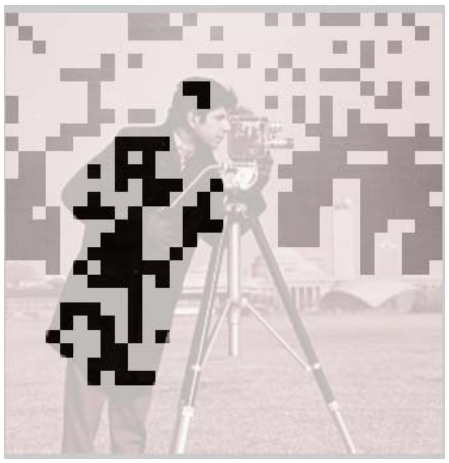
Slika 2.3: Porast entropije po blokovima.



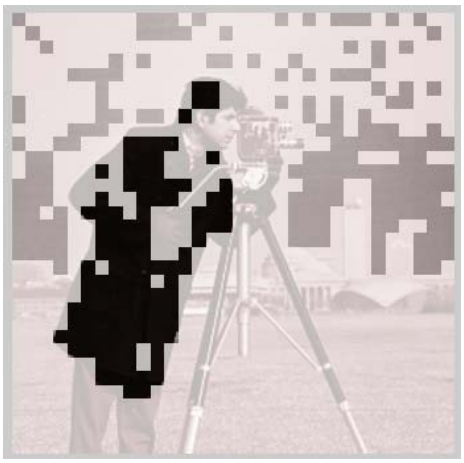
Entropija 1–2



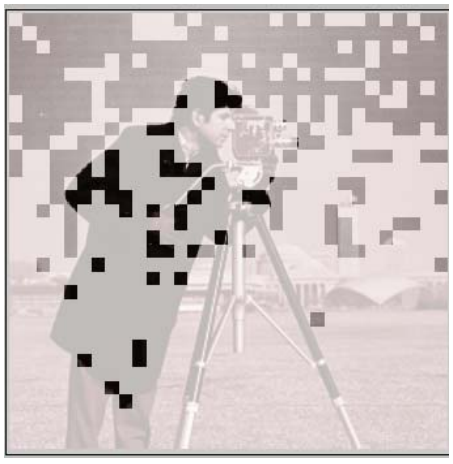
Entropija 0–2



Entropija 2-3



Entropija 0-3



Entropija 3-4

Entropija 0-4

Entropija 4-5

Entropija 0-5

Entropija 5-6

Entropija 0-6

2.3. Entropijsko kodiranje

Entropijsko kodiranje[8] je metod kompresije bez gubitaka, u kome se simbolima pridružuju kodovi, čije dužine odgovaraju verovatnoćama simbola. Entropijsko kodiranje komprimuje podatke zamenom simbola predstavljenih kodovima jednakih dužina, simbolima predstavljenim kodovima sa dužinama proporcionalnim negativnom logaritmu njihove verovatnoće. Tako, najčešći simboli koriste najkraće kodove.

Po Šenonu [2_03], optimalna dužina koda za simbol je $-\log_b P$, gde je b broj simbola izlazne azbuke, a P je verovatnoća ulaznog simbola. U slučaju binarne izlazne azbuke, optimalna dužina koda je $-\log_2 P$.

[8] Termin *kodiranje* u nazivu obuhvata obe "strane medalje", enkodiranje i dekodiranje. U opisu ove metode prvo se govori o enkodiranju, a zatim o dekodiranju simbola. Zato će se termin "enkodiranje" ovde koristiti, u cilju jasnijeg teksta (mada se u praksi može često naći i da se i taj termin zamenjuje kraćim terminom "kodiranje").

Kompresijom bez gubitaka može se postići samo umereni nivo kompresije. Po Šenonu, pri kompresiji bez gubitaka, donja granica veličine komprimovanog fajla je (za sliku dimenzije $m \times n$, entropije H):

$$H \cdot m \cdot n$$

tj. za zapis je potrebno najmanje H bitova informacije po pikselu.

Dve najviše korišćene tehnike entropijskog kodiranja su *Hafmenovo (David A. Huffman)* i *aritmetičko* kodiranje. Oba ova kodiranja uključena su u JPEG standard kompresije slika.

2.3.1. Hafmenovo kodiranje

Šenon je dokazao da se u kodiranju može dostići entropija ako se dozvoli da kašnjenje kodiranja po potrebi neograničeno raste. Entropija se dostiže kada dužina bloka koji se kodira teži beskonačnosti. Rezultati su veliki memorijski zahtev i visoka složenost izračunavanja.

U mnogim slučajevima, potreban nam je metod kodiranja koji je optimalan i trenutan za izvor informacije sa konačnim izvornim simbolima u izvornoj azbuci S. *Optimalan* znači da njegova srednja dužina predstavlja minimum među svim drugim, nad istom izvornom azbukom S i kodnom azbukom A. *Trenutan* znači da je moguće dekodirati svaku kodnu reč u sekvenci kodnih simbola bez znanja sledećih kodnih reči.

Jedan takav metod razvio je Hafmen (Huffman) 1952. godine [2_05].

Ovaj metod kompresije prevodi simbole iz izvorne azbuke

$$S = \{s_1, ..., s_m\}$$

u kodnu azbuku

$$A = \{a_1, ..., a_r\}.$$

Pri tome se svakom simbolu $s_i \in S$ pridružuje kodna reč A_i u azbuci A,

$$s_i \rightarrow A_i = (a_{i1}, ..., a_{ik})$$

gde je A_i niska k kodnih simbola. Hafmenov kôd izvornim simbolima koji se češće pojavljuju pridružuje kraće kodne reči u azbuci A.

Dužine kodnih reči $A_1, ..., A_m$ možemo označiti sa $l_1, ..., l_m$. Srednja dužina koda je

$$L_{avg} = \sum_{i=1}^{m} p(s_i) \cdot l_i$$

Po Šenonu, srednja dužina koda ograničena je odozdo entropijom izvora informacije. Entropija izvora informacije S definisana je sa:

$$H(S) = -\sum_{i=1}^{m} p(s_i) \cdot \log_2 p(s_i) \tag{2.6}$$

To znači da srednja količina koda dostiže entropiju kada je $l_i = -\log_2 p(s_i), (i = 1,...,m)$.

Efikasnost koda se definiše kao odnos entropije i srednje dužine koda: $h = \frac{H(S)}{L_{avg}}$.

Redundanca koda, z, definiše se kao $z = 1 - h$.

Hafmenova kompresija obavlja se u dva prolaza. U prvom prolazu analiziraju se podaci i na osnovu sadržaja stvara se model drveta. Drugi prolaz komprimuje podatke koristeći model stvoren u prvom prolazu.

Koraci algoritma:

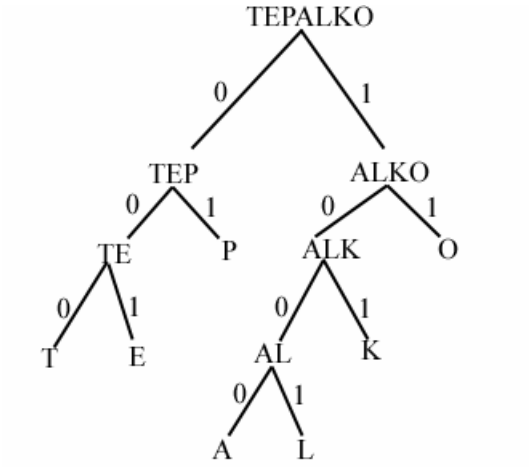
1. Simboli se uređuju tako da su verovatnoće njihovog pojavljivanja u nerastućem redosledu.
2. Kombinuju se dva najmanje verovatna izvorna simbola; tim najmanje verovatnim simbolima dodeljuju se 0 i 1; formira se novi izvorni simbol sa verovatnoćom jednakom sumi verovatnoća dva najmanje verovatna simbola.
3. Postupak se ponavlja dok se ne dođe do situacije da izvorna azbuka sadrži samo jedan izvorni simbol.
4. Da bi se pronašle odgovarajuće kodne reči, treba se, polazeći od izvornog simbola u poslednjoj pomoćnoj izvornoj azbuci, vratiti unazad ka svakom izvornom simbolu u originalnoj izvornoj azbuci

Primer 2.7: Reč POPOKATEPETL može se u azbuci $A = \{0,1\}$ kodirati na sledeći način:

P	1/4	P	1/4	P	1/4	TE	1/3	ALKO	5/12	TEP	7/12	0	TEPALKO	1
O	1/6	O	1/6	ALK	1/4	P	1/4	TE	1/3	0	ALKO	5/12	1		
T	1/6	T	1/6	O	1/6	ALK	1/4	0	P	1/4	1				
E	1/6	E	1/6	T	1/6	0	O	1/6	1						
K	1/12	AL	1/6	0	E	1/6	1								
A	1/12	0	K	1/12	1										
L	1/12	1													

Svaka od kolona prikazane tabele predstavlja jedno izvršenje koraka (1) i (2). U svakoj od kolona navedene tabele nalaze se tri "potkolone": simboli, njihove verovatnoće i dodeljene 0 i 1 najmanje verovatnim simbolima.

Posle završena prva tri koraka dobija se drvo, koje se koristi u koraku (4) (slika 2.4):



Slika 2.4: Hafmenovo kodiranje reči "POPOKATEPETL"

Kodovi se dobijaju čitanjem oznaka grana od korena do listova:

```
P    01
O    11
T    000
E    001
K    101
A    1000
L    1001
```

Tako se reč POPOKATEPETL kodira sa

P	O	P	O	K	A	T	E	P	E	T	L
01	11	01	11	101	1000	000	001	01	001	000	1001

Dužina koda je $33/12 = 2.7500$ bita po simbolu.

Entropija ove reči je

$$-(\frac{1}{4}\log_2\frac{1}{4}+3\cdot\frac{1}{6}\log_2\frac{1}{6}+3\cdot\frac{1}{12}\log_2\frac{1}{12})=\frac{1}{4}\log_2 4+\frac{1}{2}\log_2 6+\frac{1}{4}\log_2 12=$$

$$\frac{1}{2}+\frac{1}{2}+\frac{1}{2}\log_2 3+\frac{1}{2}+\frac{1}{4}\log_2 3=\frac{3}{2}+\frac{3}{4}\log_2 3\approx 2.6887,$$

što pokazuje da je dužina postignutog koda bliska entropiji.

2.3.2. Aritmetičko kodiranje

Dok Hafmenovo kodiranje svaki simbol poruke zamenjuje kodnom rečju, aritmetičko kodiranje kodira celu poruku u jedan broj, n, gde je $0 \le n \le 1$ [2_06, 2_07, 2_08].

Primer 2.8: Reč FASADA

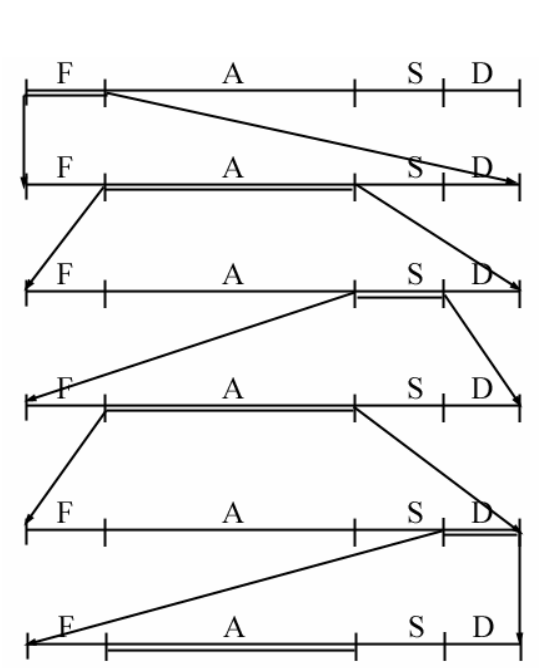
s_i	Simbol	Verovatnoća pojavljivanja $p(s_i)$	Kumulativna verovatnoća $cp(s_i)$
s_1	F	0.167	0.000
s_2	A	0.500	0.167
s_3	S	0.167	0.667
s_4	D	0.166	0.834

```
clear,close all,format compact
%      F A S A D A
nizul=[1,2,3,2,4,2];[n0,n1]=size(nizul);           %niz na ulazu
%      F    A    S    D
p=[0.167,0.500,0.167,0.166];cp=[0,0.167,0.667,0.834]; %niz verovatnoca
l(1)=cp(nizul(1));w(1)=p(nizul(1));
for i1=2:n1
    l(i1)=l(i1-1)+w(i1-1)*sum(cp(nizul(i1)));      %levi kraj intervala
    w(i1)=w(i1-1)*p(nizul(i1));                    %sirina intervala
    d(i1)=l(i1)*w(i1);                             %desni kraj intervala
end
d=l+w;l,d
```

Rezultati:

	F	A	S	A	D	A
l =	0.0000	0.0279	0.0836	0.0859	0.0917	0.0919
d =	0.1670	0.1114	0.0975	0.0929	0.0929	0.0925

Ulazna poruka je niz od n_1 karaktera: $nizul = 'FASADA'$.



Učitano slovo	Interval
F	[0.0000,0.1670)
A	[0.0279,0.1114)
S	[0.0836,0.0975)
A	[0.0859,0.0929)
D	[0.0917,0.0929)
A	[0.0919,0.0925)

Slika 2.5: Aritmetičko kodiranje reči "FASADA"

Realni interval $[0,1)$ delimo na četiri podintervala, tako da $i-$ti ima dužinu u skladu sa verovatnoćom pojavljivanja $i-$tog simbola, tj. $p(s_i), i = 1,...,4$. Sva četiri podintervala su disjunktni, jer je suma svih verovatnoća jednaka 1 (slika 2.5).

Kako je prvi simbol F, njegov podinterval je $[0,0.1670)$.

Drugi simbol je A; interval $[0,0.1670)$ se deli na 4 podintervala. Simbol A iza simbola F daje podinterval $0.167 * [0.167,0.667) = [0.0279,0.1114)$

Postupak se nastavlja. Ovde postoje dve rekurzije (rekurzija levog kraja i rekurzija širine):

$l(i_1) = l(i_1 - 1) + w(i_1 - 1) \cdot cp(nizul(i_1))$,

$w(i_1) = w(i_1 - 1) \cdot cp(nizul(i_1))$

$cp(nizul(i_1))$ je kumulativna verovatnoća u novoj rekurziji. Interval za $i_1 - $ti simbol je $[l(i_1), d(i_1))$, gde je $d(i_1) = l(i_1) + w(i_1)$

Ulaz u program je niska izvornih simbola; izlaz je podinterval intervala [0,1). Teoretski, svaki realni broj u izlaznom intervalu može biti proglašen za izlaz ulazne niske simbola, jer su svi podintervali disjunktni. Ipak, uobičajeno je da se kao izlazna vrednost koristi levi kraj izlaznog intervala.

Dekoder zna proceduru enkodiranja, i zato ima informacije o rasporedu podintervala. On zato upoređuje levi kraj finalnog intervala sa svim krajnjim tačkama. Tako zna koji je interval dodeljen prvom simbolu.

Oduzima levi kraj podintervala koji odgovara tom prvom simbolu, od levog kraja finalnog intervala. Broj koji dobije, deli širinom podintervala koji odgovara prvom simbolu.

Vidi se da su navedeni koraci ustvari "*undo*" od operacija procedure enkodiranja.

Enkodiranje i dekodiranje iziskuju samo aritmetičke operacije (otud i naziv "aritmetičko kodiranje"): sabiranje i množenje u enkodiranju, a oduzimanje i deljenje u dekodiranju.

Enkodiranje se završava stavljanjem simbola završetka na kraju niske ulaznih simbola. Tako će sistem aritmetičkog kodiranja znati kada da završi sa dekodiranjem.

Širina finalnog intervala postaje sve manja, ako niska ulaznih simbola raste. To dovodi do *problema preciznosti*. Taj problem, koji dugo nije dozvoljavao da se aritmetičko kodiranje praktično koristi, rešen je kasnih osamdesetih godina prošlog veka.

Sa rastućim brojem izvornih simbola, levi i desni kraj intervala postaju sve bliži. Ono što u se u navedenom primeru vidi je da je posle pročitana tri znaka (FAS), poznato da je prva cifra iza decimalne tačke 0; posle pročitanih prvih pet znakova (FASAD), poznata je sledeća cifra: 9.

U trenutku kada se zna sledeća cifra, ne mora se više zadržavati – može biti poslata (i pri tome je više ne treba pamtiti), bez uticaja na finalni rezultat. Tako je moguće postupno izdavati izlaz (čim se dođe do sledeće cifre) i primati ulaz.

Jasno, u računaru postoji razlika u odnosu na prikazan algoritam. Po prirodi stvari, koristiće se binarne cifre i binarna aritmetika, ali razlike nisu suštinske, i opisani algoritam važi i dalje.

2.4. Neentropijska kodiranja

Entropijsko kodiranje ima prednost u odnosu na neentropijsko u slučaju da je izvor podataka "bez memorije". Ako se očekuje neka pravilnost u podacima (duge sekvence istog simbola ili često ponavljanje pojedinih sekvenci), RLE i rečnički algoritmi mogu dovesti do manjih fajlova.

2.4.1. RLE kodiranje

RLE (Run Length Encoding) [2_09] je vrlo jednostavan oblik kompresije u kome se sekvenca sastavljena od n pojavljivanja istog simbola zamenjuje samo jednim tim simbolom i brojem n (brojem simbola u tom nizu). Tako, vrednosti piksela u bitmapiranoj slici koja se sastoji samo iz crnih (B) i belih (W) piksela, na primer $WWWWWWWWWWBBWWWWWBWWWWWWWWW$ mogu se zameniti sa $10W2B5W1B9W$.

Na taj način, originalnih 27 karaktera zamenjuje se sa 11. Naravno, stvarni format korišćen za čuvanje slika sastoji se od binarnih, a ne kao ovde, od ASCII karaktera, ali je princip isti.

Ovaj metod je koristan za kompresiju binarnih (crno–belih) slika (kao što su skenovi crnog teksta na belom papiru), a i za crteže ili slike sa malo boja (to su pre svega paletne slike). Ne radi dobro na slikama sa kontinualnim bojama, kao što su fotografije.

2.4.2. Rečničko kodiranje

Ovi algoritmi komprimuju podatke fajla, tako što određene grupe simbola zamenjuju jednim kodom. Primer takvog algoritma bio bi kada bi se svakoj reči u jeziku dodelio kôd. Ako jezik ima 100000 reči, svaka od njih bi se mogla predstaviti sa 17 bitova (jer $2^{17} > 100000$). Tako bi za svaku reč bilo dovoljno nešto više od dva bajta, što jeste manje nego da se za svako slovo upotrebi sedam ili osam bitova (što sugeriše ASCII zapis). Ovo rešenje ima za osnovnu manu što se dopuštaju samo tih 100000 reči – promena jezika ili pojava reči izvan kodiranog skupa učinili bi da algoritam ne funkcioniše. Takođe, da bi se kodiranje moglo obaviti, mora se posedovati ceo rečnik. Naravno, za većinu reči iz rečnika nije baš verovatno da će se pojaviti; manji skup reči dao bi efikasniji kôd – sa većom uštedom u prostoru.

Sve ovo sugeriše da je bolje rešenje da se rečnik pravi adaptivno, za podatke fajla. Primer takve kompresije je LZW kompresija (ime je dobila po autorima – Lempel, Ziv, Welch) [2_10, 2_11, 2_12].

Mnogi fajlovi, posebno tekst fajlovi, imaju određene niske koje se često pojavljuju.

LZW polazi od "rečnika" koji se sastoji od reči dužine 1 (karakteri sa indeksima 0–255). On dalje proširuje rečnik tokom čitanja informacije.

Program čita karakter po karakter. Ako kôd postoji u rečniku, dodaje ga u tekuću radnu nisku, i čita sledeći karakter.

Ako radna niska nije u rečniku, dodaje je u rečnik i šalje dalje radnu nisku bez novog karaktera. Zatim postavlja radnu nisku kao novi karakter.

3. Kompresija sa gubicima

Kompresijom sa gubicima postižu se znatno veće uštede u prostoru, nego što je to moguće kompresijom bez gubitaka. Tu se pre svega koriste osobine ljudskog oka koje nekim informacijama pridaje veći značaj nego drugim. Odbacujući vizuelno manje važnu informaciju, podaci mogu biti komprimovani znatno intenzivnije.

Slika rekonstruisana posle kompresije sa gubicima, ima oštećenja u odnosu na original. Međutim, pod normalnim uslovima gledanja ta oštećenja nisu primetna (ova kompresija je *vizuelno bez gubitaka*).

Nešto kasnije, u poglavlju o vernosti i kvalitetu digitalne slike u koju je ugrađen žig, biće detaljnije razmatrana ljudska psihovizuelna percepcija i svojstva ljudskog vizuelnog sistema (*human visual system*, HVS). Poznavanje svojstava HVS pomaže da se u slikama otkrije *psihovizuelna redundanca*.[9] Uklanjanjem ove redundance, dobiće se slika koju je moguće smestiti u manji prostor nego što je to slučaj sa originalnom slikom.

Nekoliko takvih osobina HVS:

- Ljudsko oko je osetljivije na signal sjajnosti nego na signal boje.
- Ljudsko oko je osetljivije na komponente niže nego na komponente više prostorne frekvencije.
- Ljudsko oko je osetljivije na promene sjajnosti u tamnijim nego u svetlijim oblastima slike.

Metode kompresije sa gubicima mogu se podeliti u nekoliko grupa:

- *Redukovanje prostora boja* na najviše korišćene boje na slici. Odabrane boje se zadaju u *kolor paleti*, u zaglavlju komprimovane slike. Boja svakog piksela zadaje se indeksom na boju u paleti. Ovaj metod može se kombinovati sa

[9] Neki algoritmi digitalnog vodenog žiga koriste ova znanja tako što će se većina podataka žiga kodirati baš u toj redundanci (time je žig manje primetan).

diterovanjem[10] da bi se stvorio utisak većeg broja različitih boja, i promena boja izgledala manje nagla.

- *Chroma subsampling* – korišćenje manje rezolucije za boju (chroma) nego za sjajnost (*intenzitet, luma*). Koristi se činjenica da oko zapaža promene u sjajnosti bolje nego promene u boji. Zato se iz slike, radi uštede prostora, odbacije polovina, pa i više, informacije o boji.
- *Transform coding*. Na sliku u nijansama sive, primenjuje se neka transformacija (najčešće je to diskretna kosinusna ili neka od transformacija talasićima), a zatim i kvantizacija i entropijsko kodiranje.
- *Kompresija fraktalima*.

U praksi se često neke od ovih metoda kombinuju. Tako, na primer, JPEG kompresija za sliku u boji obuhvata u sebi dve metode (chroma subsampling i transform coding).

Za upoznavanje sa sadržajem ovog poglavlja korisne su knjige [3_01, 3_02].

3.1. JPEG kompresija

Ovo je vrlo moćna tehnika kompresije sa gubicima [3_03, 3_04]. Pre svega je namenjena kompresiji fotografija u punoj boji. U stanju je da bez (za ljudsko oko) vidljivih gubitaka u kvalitetu slike smanji veličinu fajla na njen, recimo, dvadeseti deo. Ovo u odnosu na tehnike kompresije bez gubitaka sa kojima se obično memorijski prostor smanjuje za jedva 50%, predstavlja veliku uštedu.

JPEG najbolje komprimuje fotografije u punoj boji. Slabije, ali ipak dosta dobro, pokazuje se kod slika u nijansama sive. Najslabiji je kod jednostavnih crteža sa oštrim ivicama, gde se i pri manjem intenzitetu kompresije pojavljuju vidljivi defekti – artifakti.[11]

---

[10] *Diterovanje* je simulacija više boja u paleti. U monohromatskom sistemu, koji prikazuje ili štampa samo crno i belo, nijanse sive je moguće simulirati stvaranjem različitih obrazaca crnih tačaka. U kolor sistemima, dodatne boje mogu se simulirati variranjem obrazaca tačaka postojećim bojama. Diterovanje ne može da proizvede sasvim iste rezultate kao sa potrebnom dubinom boja, ali može da učini da osenčeni crteži i fotografije izgledaju realističnije. U praksi se često slika u punoj boji diteruje na 256 boja. Engleska reč *dither* (u rečniku: *oklevanje*) vodi poreklo od srednjoengleske reči *didderen*, što znači *podrhtavati*.

[11] Pojam *artifakt* (ili *artefakt*) u osnovi označava objekat nastao iz ljudske aktivnosti u određenom istorijskom periodu. Tako artifaktom nazivamo fizički objekat u biblioteci, arhivu ili muzeju.

Ovaj pojam se danas koristi sa velikim brojem različitih značenja, u zavisnosti od oblasti o kojima se govori. *Kompresioni artifakt* kod slike je rezultat agresivne kompresije podataka slike, kod koje se uništavaju podaci za koje se smatra da su manje važnosti od ostalih podataka, ali koji su ipak primetni i neprijatni za posmatrača.

Korisnik može da se, u skladu sa namenom slike, opredeli za nivo kompresije. Što je viši nivo kompresije, fajlovi će biti manji, a oštećenja slike veća.

JPEG standard kompresije danas postoji u različitim verzijama. Generalno, ova tehnika se za slike dubine boje 24 (po 8 bitova za crvenu, zelenu i plavu komponentu) sastoji iz sledećih koraka.

(1) Transformacija prostora boja: Podaci slike konvertuju se iz RGB u YCbCr prostor boja. *Prostor boja* je prostor u kome je svaka boja određena svojim koordinatama. Postoji mnogo načina da se prostor boja definiše. Primeri su prostori CMY & CMYK, IHS, HSL, HSV, Lab (L*a*b*), RGB, YCbCr,YIQ.

U RGB prostoru, koordinate su određene komponentama crvene, zelene i plave boje. U prostoru YCbCr, komponenta Y predstavlja sjajnost (luminance, brightness) piksela, a komponente Cb i Cr zajedno predstavljaju komponente boje (chrominance). Promena prostora boja korisna je zato što ljudsko oko vidi više detalja u Y komponenti nego u druge dve; konverzija omogućuje da komponente Cb i Cr budu komprimovane grublje od komponente Y.

Za ovo postoji i drugo tumačenje. Komponenta Y predstavlja informaciju o sjajnosti, a preostale dve komponente – kolor informaciju. **Oko koristi ivice – granice oblasti sjajnosti da prepozna granice objekata**, a informacija o nijansi boje za njega nije ni blizu toliko bitna.

Cb i Cr komponente se redukuju tako što se smanjuje dimenzija slike u ove dve komponente. To je malopre spomenuta chroma subsampling (koristi se i termin *downsampling*). Tako se smanjuje veličina u bajtovima i za celu sliku. Ovo je jedan od koraka JPEG kompresije u kome se neki podaci nepovratno gube. Zbog ovog koraka postiže se viša mera kompresije za JPEG slike u boji, nego za slike u nijansama sive.

U ostatku procesa kompresije, Y, Cb i Cr podslike originalne slike obrađuju se zasebno, na međusobno vrlo sličan način.

(2) Diskretna kosinusna transformacija (Discrete Cosine Transform, DCT): Svaka komponenta slike (Y, Cb, Cr) deli se u blokove od po 8×8 piksela, pa se svaki blok konvertuje u prostor frekvencije koristeći dvodimenzionu diskretnu kosinusnu transformaciju (DCT) [3_05, 3_06]. Ako dimenzije slike nisu multipli broja 8, nekompletni blokovi – oni u desnom i donjem delu slike se prethodno dopunjavaju "lažnim" podacima (bilo crnim pikselima, bilo ponavljanjam piksela sa ivice slike).

Dvodimenziona DCT i inverzna dvodimenziona DCT matrice S dimenzije $m \times n$ date su sa:

$$D(u,v) = \frac{2}{\sqrt{mn}} C(u)C(v) \sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y) \cos \frac{\pi (u-1)(2x-1)}{2m} \cos \frac{\pi (v-1)(2y-1)}{2n},$$

$$u = 1,...,m,\ v = 1,...,n \tag{3.1}$$

$$S(x,y) = \frac{2}{\sqrt{mn}} \sum_{u=1}^{m} \sum_{v=1}^{n} C(u)C(v)D(u,v) \cos \frac{\pi (u-1)(2x-1)}{2m} \cos \frac{\pi (v-1)(2y-1)}{2n},$$

$$x = 1,...,m,\ \ y = 1,...,n\,, \tag{3.2}$$

gde $C(u) = 1/\sqrt{2}$ ako $u = 1$, inače $C(u) = 1$,

uz primedbu da je kod JPEG-a $m = n = 8$.

Koeficijenti matrice D (matrica slike u domenu transformacije) dele se na "DC koeficijent" i "AC koeficijente". DC je prvi koeficijent: $D(1,1)$. On ima nula frekvenciju u obe dimenzije. AC su preostalih (tj. $mn - 1$) koeficijenata sa nenula frekvencijama.

DC koeficijent matrice jednak je

- sumi svih elemenata matrice podeljenoj korenom proizvoda dimenzija,
- srednjoj vrednosti elemenata matrice pomnoženoj korenom proizvoda dimenzija,
- kvadratnom korenu proizvoda sume i srednje vrednosti elemenata matrice.

DCT korak koncentriše većinu signala (magnitude elemenata su znatno više) u nižim prostornim frekvencijama (gornji levi ugao matrice D).

(3) Kvantizacija: *Kvantizacija* se definiše kao proces u kojem se uzorkuje neprekidan opseg vrednosti analognog signala i deli u nepreklapajuće (ali ne obavezno jednake) podoblasti, i diskretna, jedinstvena vrednost dodeljuje svakoj podoblasti.

Ljudsko oko dosta slabo razlikuje promene sjajnosti na visokim frekvencijama (tj. na malim rastojanjima). Ova činjenica omogućuje da se odbaci znatna količina informacije u komponentama visoke frekvencije. To se obavlja kvantizacijom – prostim deljenjem svake komponente u domenu frekvencije sa konstantom određenom u tabeli kvantizacije za tu komponentu, i zatim zaokruživanjem na najbliži ceo broj (u skladu sa faktorom kvaliteta, vrednosti iz ove tabele mogu se množiti i nekom konstantom). Tabela koja sledi je kvantizaciona tabela za komponentu sjajnosti (Y) prema predlogu *IJG* [12].

---

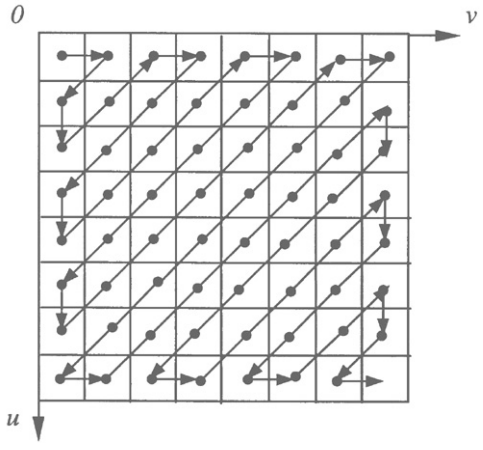
[12] *Independent JPEG Group*, neformalna grupa koja piše i distribuira slobodnu biblioteku za JPEG kompresiju slika.

Kvantizacija je deo algoritma JPEG kompresije u kome se najviše informacije gubi. Kao rezultat ove operacije, mnoge od komponenti visoke frekvencije zaokružuju se na nulu, ili postaju mali pozitivni ili negativni brojevi.

(u,v)	1	2	3	4	5	6	7	8
1	16	11	10	16	24	40	51	61
2	12	12	14	19	26	58	60	55
3	14	13	16	24	40	57	69	56
4	14	17	22	29	51	87	80	62
5	18	22	37	56	68	109	103	77
6	24	35	55	64	81	104	113	92
7	49	64	78	87	103	121	120	101
8	72	92	95	98	112	100	103	99

Tabela 3.1: Vrednosti kvantizacije korišćene u JPEG kompresiji (komponenta Y)

(4) Entropijsko kodiranje: Kvantizovane komponente slike uređuju se u "cikcak" redosled (slika 3.1), čime se dodatno grupišu slične frekvencije i povećavaju dužine sekvenci nula. Zatim se koristi Hafmenovo kodiranje tako uređenog niza. JPEG standard dozvoljava i korišćenje aritmetičkog kodiranja umesto Hafmenovog. Aritmetičko kodiranje je matematički superiornije od Hafmenovog, ali se retko koristi zato što je pokriveno patentima, zato što su (en)kodiranje i dekodiranje dosta sporiji, i zato što nije "mnogo" nego "samo malo" bolje (fajlovi su obično oko 5% manji).



Slika 3.1: Cikcak sken DCT koeficijenata u 8×8 bloku.

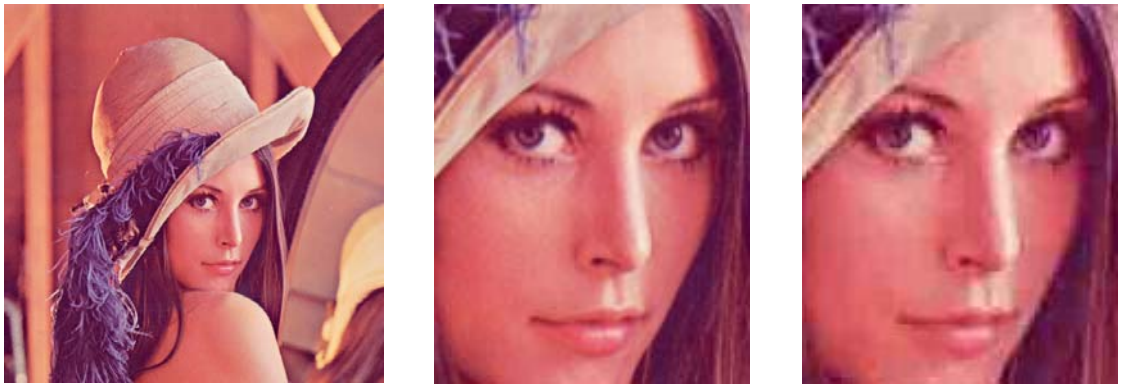
Dekompresija

Da bi se slika prikazala, mora se dekomprimovati. Koraci dekompresije su obrnuti koracima kompresije. U procesu JPEG dekompresije svi DCT koeficijenti se *dekvantizuju* (tj. množe vrednostima kvantizacije koje su korišćene u kompresiji). Zatim se na rekonstruisane podatke primenjuje inverzna DCT. Restaurirana slika će biti bliska (ali ne i identična) sa originalom; ali, ako se sa kompresijom nije preteralo, razlike neće biti primetne za ljudsko oko.

Mera kompresije i artifakti

Rezultujuća mera kompresije može se menjati u skladu sa izborom delilaca korišćenih pri kvantizaciji. Kompresija 10:1 obično rezultuje u slici koju ljudsko oko ne može da razlikuje od originala. Kompresija od 100:1, mada obično moguća, imaće uočljive blokovske artifakte. Ovi kompresioni artifakti pojavljuju se zbog koraka kvantizacije JPEG algoritma.

Na slici 3.2 redom su date originalna TIF i uvećani deo (lice) TIF i JPEG komprimovane 'Lena slike' ('Lenna image'). Na uvećanom delu JPEG slike uočljivi su artifakti nastali kompresijom (veličina TIF slike je 780 KB, a JPEG – 28 KB, što predstavlja kompresiju od oko 28:1).



Slika 3.2: 'Lena slika': TIF, i detalj – TIF i JPEG (kompresija 28:1)

'Lenna image' je inače verovatno najpoznatija među svim slikama koje se u svetu koriste u obradi slika. Ona predstavlja deo veće slike koja se pojavila u časopisu "Playboy", kao duplerica za novembar 1972. Devojka je Šveđanka Lena Sjööblom. Na originalnoj, većoj slici[13], u donjem desnom uglu je poruka da je nosilac autorskih prava nad ovom slikom Playboy. Međutim, neko je sliku opsekao i zadržao samo njen manji deo; kasniji

[13] Cela slika se može naći na adresi http://www.lenna.org/full/len_full.html

korisnici koji su je ovako opsečenu uzeli sa Interneta, obično i ne znaju da koriste sliku čiji je vlasnik poznat.

3.2. Transformacije slike

Centralni deo kompresije sa gubicima je transformacija (transform), ili preslikavanje podataka slike iz prostornog domena, u domen transformacije (frekvencije). Transformacije su uglavnom unitarne (obično i ortogonalne), pa se njima čuva energija slike. I ne samo to: energija u domenu transformacije skoncentrisana je na sasvim malom broju koeficijenata. O energiji slike biće reči nešto kasnije.

DCT nije jedina takva transformacija. U literaturi su opisane i neke druge, kao: diskretna Fourier-ova (DFT), diskretna Walsh–Hadamard-ova (WHT), diskretna Karhunen–Loeve (KLT), ... Među pobrojanim, najbolje rezultate u kompresiji ima KLT, ali je najkomplikovanija za korišćenje, jer iziskuje da se matrica transformacije računa za svaku posebnu sliku. Ostale pobrojane transformacije – DCT, DFT, WHT – koriste matricu transformacije koja zavisi od dimenzija slike, ali ne i od same slike. Među njima, izvršenje najbliže izvršenju KLT ima DCT. DFT je ipak, među pobrojanim transformacijama najpoznatija [3_01, 3_05, 3_12]].

3.2.1. DCT i blok DCT – slikovni prikaz

Diskretna kosinusna transformacija (DCT) matrice dimenzije $m \times n$ može se dobiti pomoću formule [1_07]

$$D = M_m^{-1} \cdot S \cdot M_n \tag{3.3}$$

S i D su matrice slike dimenzije $m \times n$ (S – u prostornom, a D – u domenu transformacije)

M_m i M_n su kvadratne matrice dimenzija $m \times m$, odnosno $n \times n$, i mogu biti izračunate pomoću sledećeg koda (ovde ispisan za matricu dimenzije $m \times m$, označenu sa M):

```
c=ones(m)*sqrt(2/m); c(1)=sqrt(1/m);
for i=1:m
    for j=1:m
        M(i,j)=c(j)*cos((2*i-1)*(j-1)*pi/(2*m));
    end
end
```

Ove matrice su konstante – ne zavise od sadržaja slike S, nego samo od njene dimenzije. One su i **ortogonalne** – važi $M_m^{-1} = M_m^T \quad (m \in N)$

Sledi prikaz nekoliko takvih matrica: M_2, M_3, M_4, M_8 i M_{128} (za diskretnu kosinusnu transformaciju):

M_2

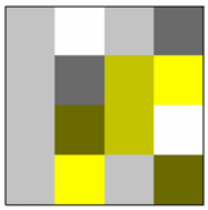
```
0.707     0.707
0.707    -0.707
```


M_3

```
0.577     0.707     0.408
0.577     0.000    -0.816
0.577    -0.707     0.408
```

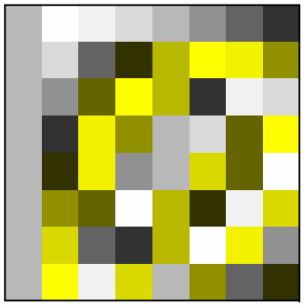
M_4

```
0.500  0.653  0.500  0.271
0.500  0.271 -0.500 -0.653
0.500 -0.271 -0.500  0.653
0.500 -0.653  0.500 -0.271
```

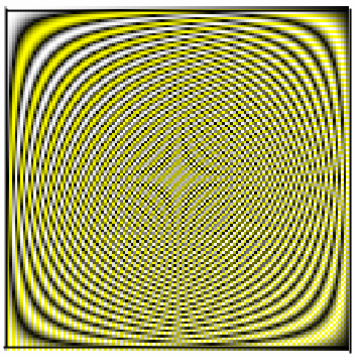


M_8

```
0.35  0.49  0.46  0.42  0.35  0.28  0.19  0.10
0.35  0.42  0.19 -0.10 -0.35 -0.49 -0.46 -0.28
0.35  0.28 -0.19 -0.49 -0.35  0.10  0.46  0.42
0.35  0.10 -0.46 -0.28  0.35  0.42 -0.19 -0.49
0.35 -0.10 -0.46  0.28  0.35 -0.42 -0.19  0.49
0.35 -0.28 -0.19  0.49 -0.35 -0.10  0.46 -0.42
0.35 -0.42  0.19  0.10 -0.35  0.49 -0.46  0.28
0.35 -0.49  0.46 -0.42  0.35 -0.28  0.19 -0.10
```



M_{128}



Diskretna kosinusna transformacija **prevodi matricu slike u drugi koordinatni sistem**. Naime, slika S u originalnom sistemu (u tzv. *prostornom domenu*) zadata je baznim slikama $B_s^{i,j}$, ($i = 1,2,...,m$, $j = 1,2,...,n$),

$B_s^{i,j}(i_1, j_1) = 1$ za $i_1 = i$ i $j_1 = j$

$B_s^{i,j}(i_1, j_1) = 0$ za $i_1 \neq i$ ili $j_1 \neq j$

$$S = (S_{i,j})_{i=1,\ldots,m;j=1,\ldots,n} = \sum_{i=1}^{m}\sum_{j=1}^{n} S_{i,j} \cdot B_s^{\,i,j} = \sum_{i=1}^{m}\sum_{j=1}^{n} D_{i,j} \cdot B_d^{\,i,j} \qquad (3.4)$$

Matrica $D = (D_{i,j})_{i=1,\ldots,m;j=1,\ldots,n}$ predstavlja sliku u tzv. *domenu transformacije* (*domenu frekvencije*). Ovde je matrica slike predstavljena u bazi zadatoj baznim slikama $B_d^{\,i,j}$, ($i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$).

Matrice S i D se mogu posmatrati kao prikazi iste slike u dve različite baze.

Primer 3.1: Ako je slika u prostornom domenu predstavljena matricom $S = \begin{bmatrix} 0 & 50 \\ 100 & 200 \end{bmatrix}$, tada je $\left(\text{pošto } M_2 = M_2^{-1} = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\right)$,

$$D = M_2^{-1} \cdot S \cdot M_2 = \begin{bmatrix} 175 & -75 \\ -125 & 25 \end{bmatrix}.$$

U prostornom domenu, slika je predstavljena u standardnoj bazi

$B_s = \{b_s^{11}, b_s^{12}, b_s^{21}, b_s^{22}\}$

$$b_s^{11} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ b_s^{12} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ b_s^{21} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},\ b_s^{22} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

sa

$$\boldsymbol{S} = 0 \cdot \boldsymbol{b_s}^{11} + 50 \cdot \boldsymbol{b_s}^{12} + 100 \cdot \boldsymbol{b_s}^{21} + 200 \cdot \boldsymbol{b_s}^{22}$$

U domenu trasformacije, ista slika predstavljena je u bazi $B_d = \{b_d^{11}, b_d^{12}, b_d^{21}, b_d^{22}\}$, gde je $b_d^{11} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $b_d^{12} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$, $b_d^{21} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$, $b_d^{22} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$,

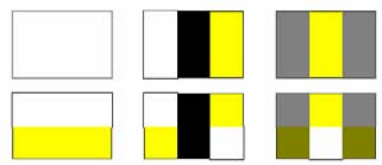
matricom

$$D = 175 \cdot b_d^{11} - 75 \cdot b_d^{12} - 125 \cdot b_d^{21} + 25 \cdot b_d^{22}$$

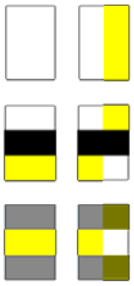
Sledi prikaz baznih slika B_s i B_d za matrice dimenzije 2×2 (slika 3.3).

Slika 3.3: Bazne slike B_s (levo) i B_d (desno) za matrice dimenzije 2×2

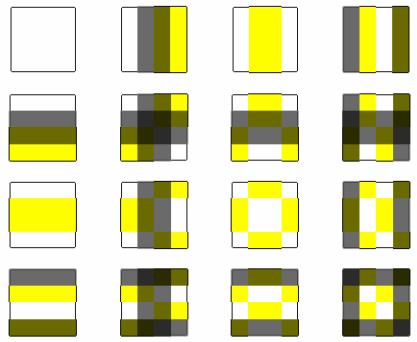
Niže su (slike 3.4, 3.5, 3.6 i 3.7) predstavljene bazne slike B_d za još neke dimenzije matrica.



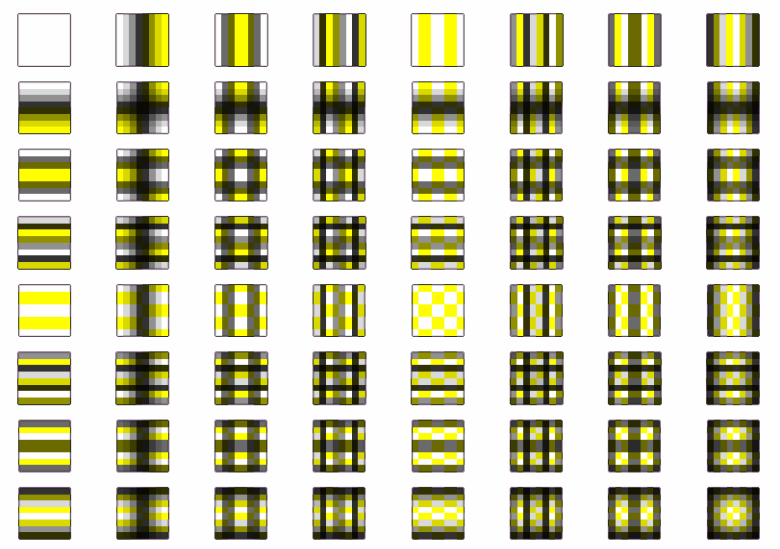
Slika 3.4: Bazne slike za dimenzije 2×3



Slika 3.5: Bazne slike za dimenzije 3×2



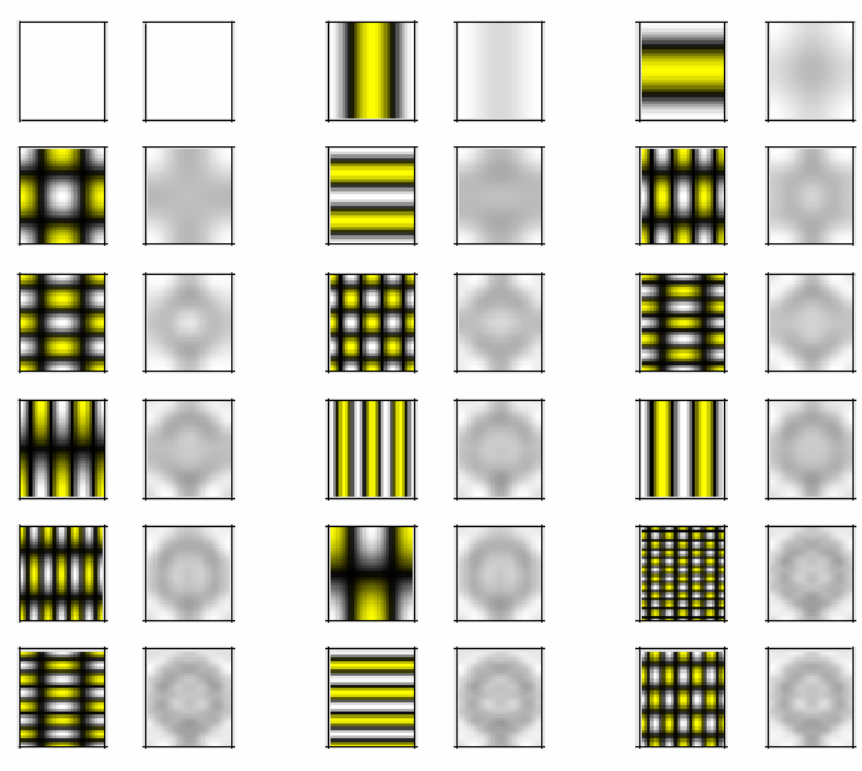
Slika 3.6: Bazne slike za dimenzije 4×4



Slika 3.7: Bazne slike za dimenzije 8×8

Primer 3.2*:* Nekoliko prvih koraka u predstavljanju slike grba Matematičkog fakulteta, Beograd mogu se videti na slici 3.8:

Originalna slika:



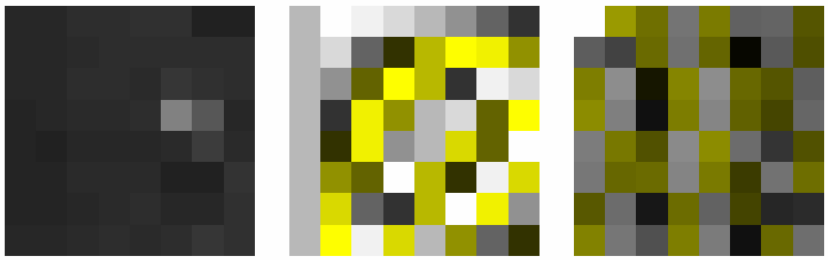
Slika 3.8: Predstavljanje slike u bazi B_d

Originalna slika ima dimenziju 32×28, i predstavlja se u bazi B_d linearnom kombinacijom 896 baznih slika (svaka od njih se množi odgovarajućim elementom u matrici D). Ovde je prikazano prvih 18 koraka u izvođenju originalne iz baznih slika. Svaki korak predstavljen je parom (bazna slika, rezultujuća slika). Koraci su uređeni u skladu sa opadajućim redosledom magnituda koeficijenata u matrici D.

Ilustracija blok DCT i JPEG kompresije: JPEG kompresija slike (u nijansama sive) S zasniva se na blokovskoj DCT:

1. Slika S deli se na blokove slike dimenzija 8×8 piksela
2. Na svaki blok slike primenjuje se DCT. Rezultat je matrica D, koja nastaje spajanjem svih 8×8 blokova u domenu transformacije.
3. Svaki element svakog 8×8 bloka izlaže se kvantizaciji. Ovo je "lossy" korak u kompresiji – on omogućuje da se podaci slike mogu čuvati u manje memorijskog prostora, u odnosu na podatke originalne slike.

Primer 3.3: Sledi (slika 3.9) prikaz za jedan 8×8 blok matrice S – S_8, M_8 i D_8 ($D_8 = M_8^{-1} \cdot S_8 \cdot M_8$):



Slika 3.9: S_8, M_8 **i** D_8

Blok D_8 – rezultat diskretne kosinusne transformacije (takođe dimenzije 8×8), za razliku od bloka u prostornom domenu, ima značajnu razliku između vrednosti u različitim delovima bloka. Vrednosti njegovih elemenata u blizini gornjeg levog ugla (tj. DC elementa) imaju daleko veće magnitude od elemenata u ostatku bloka. Ove vrednosti opadaju, idući ka donjem desnom uglu. Rezultujuća matrica D dobija se spajanjem svih blokova D_8 slike.

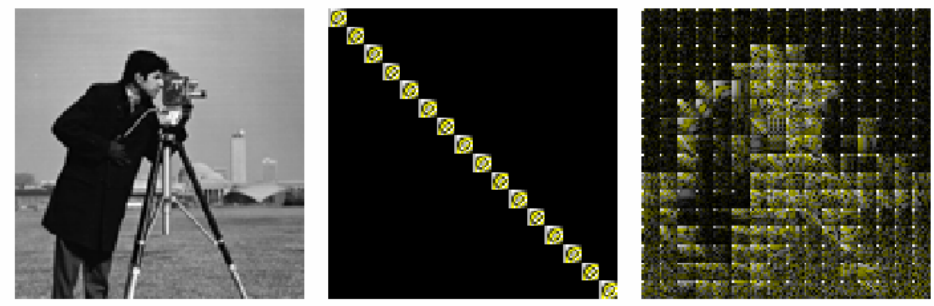
Isti rezultat kao pri povezivanju svih 8×8 DCT blokova, dobićemo množenjem (cele) matrice S slike dimenzije $m\times n$ (m i n su multipli broja 8), kvadratnim matricama ${M_{c\_m}}^{-1}$ (dimenzije $m\times m$) M_{c_n} (dimenzije $n\times n$) – rezultat je opet matrica D :

$$D = {M_{c\_m}}^{-1}\cdot S\cdot M_{c\_n} \qquad (3.5)$$

Matrica M_{c_m} (takođe i M_{c_n}) je skoro dijagonalna. Na svojoj dijagonali ima matrice M_8, a izvan nje su joj svi elementi nula. Jasno, i ona je ortogonalna (${M_{c\_m}}^{-1} = {M_{c\_m}}^{T}$).

Na slici 3.10 prikazan je primer – za sliku 'Cameraman', dimenzije 128×128 :

$D = {M_{c\_128}}^{-1}\cdot S\cdot M_{c\_128}$.



Slika 3.10: S **(levo),** M_{C_128} **(u sredini),** D **(desno) za sliku 'Cameraman'**

U svakom 8×8 bloku D_8 matrice D, elementi blizu DC elementa imaju veće magnitude u odnosu na ostatak bloka; s druge strane, oni imaju manje kvantizacione koeficijente u odnosu na udaljene elemente. Tako, kvantizacija je intenzivnija (više informacije se gubi) u elementima bloka udaljenim od DC elementa.

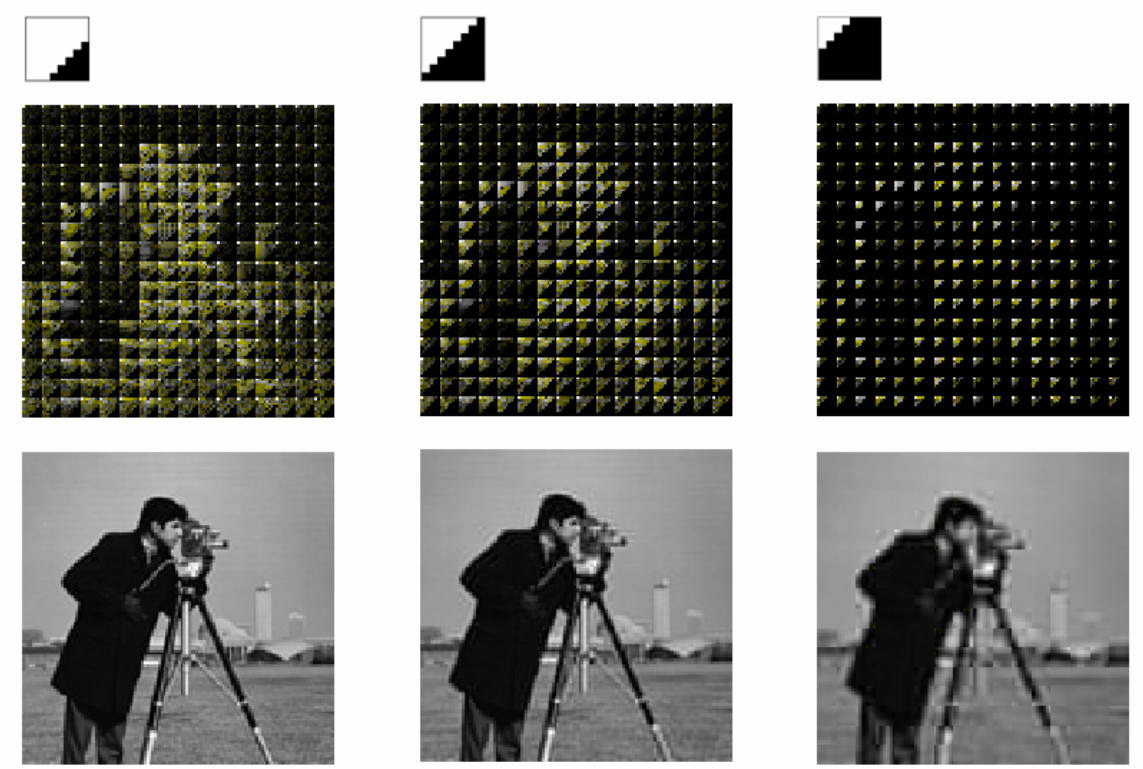
Zato, u JPEG kompresiji, podaci u donjem desnom uglu svakog DCT bloka se daleko više gube nego oni u gornjem levom uglu.

Primer 3.4: Na slici 3.11 pokazan je uticaj anuliranja podataka[14] u donjim desnim delovima blokova D_8 matrice D (efekat anuliranja za mali deo svakog 8×8 bloka; za polovinu svakog bloka; za veći deo svakog bloka slike u domenu transformacije).

Svaki od prikazana tri primera predstavljen je sa tri slike:

- oznakom dela bloka koji će biti anuliran (gornji red);
- matricom D' koja je nastala iz D anuliranjem delova koeficijenata u 8×8 blokovima;
- matricom S' koja je nastala iz D' primenom inverzne DCT.

Inverzna DCT (množenje inverzom matrice M_{c_128}) vraća matricu u prostorni domen – slika je u izvesnoj meri izobličena. Ako se nije preteralo sa anuliranjem, ove promene neće biti primetne za ljudsko oko.



Slika 3.11: Ilustracija rezultata anuliranja dela koeficijenata u domenu (blok DCT) transformacije

[14] U ovim primerima, u svrhu ilustracije, komplikovanija (a ne mnogo drugačija) kvantizacija zamenjena je sa *anuliranjem* podataka u donjem desnom uglu svakog bloka.

3.2.2. Diskretna Furijeova transformacija (DFT)

Diskretna (dvodimenziona) Furijeova transformacija (discrete Fourier transform, DFT) [3_07]:

$$F(u,v)=\sum_{x=1}^{m}\sum_{y=1}^{n}S(x,y)\cdot e^{-2i\pi\cdot(u-1)\cdot(x-1)/m}\cdot e^{-2i\pi\cdot(v-1)\cdot(y-1)/n},\quad u=1,...,m,\ v=1,...,n \tag{3.6}$$

Inverzna diskretna dvodimenziona Furijeova transformacija (IDFT) data je sa

$$S(x,y)=\frac{1}{mn}\sum_{u=1}^{m}\sum_{v=1}^{n}F(u,v)\cdot e^{2i\pi\cdot(u-1)\cdot(x-1)/m}\cdot e^{2i\pi\cdot(v-1)\cdot(y-1)/n},\quad x=1,...,m,\ y=1,...,n \tag{3.7}$$

Faktor normalizacije koji množi sumu (ovde 1 u direktnoj i $1/mn$ u inverznoj transformaciji) i znak eksponenta su samo konvencije, i razlikuju se u nekim realizacijama. Ono što je svuda isto je da direktna i inverzna transformacija imaju eksponente suprotnih znakova i da je proizvod njihovih normalizacionih faktora $1/mn$. Normalizacija od $1/\sqrt{mn}$ za DFT i IDFT čini transformacije unitarnim, ali je često praktičnije u numeričkom računanju da se svo skaliranje obavi odjednom.

Diskretna Furijeova transformacija (DFT) matrice S dimenzije $m\times n$ može se dobiti korišćenjem formule [1_07]

$$F=M_m^{-1}\cdot S\cdot M_n \tag{3.8}$$

S je matrica originalne slike (u prostornom domenu), a F – u domenu DFT; obe su dimenzije $m\times n$. Kvadratne matrice M_m i M_n imaju dimenzije $m\times m$ odnosno $n\times n$, i mogu biti izračunate pomoću *Matlab* koda (matrica je ovde označena sa M, i dimenzija je $m\times m$):

```
for i1=1:m
    for j1=1:m
        M(i1,j1)=exp(-2*pi*i/m)*(i1-1)*(j1-1);
    end
end
```

F i M_m (i M_n) su kompleksne matrice.

Matrice $M_m\ (m\in N)$ su **simetrične** – važi $M^T=M$. One su i **unitarne** – važi $M_m^{-1}=M_m^*$. Sledi prikaz nekoliko M_m matrica za Furijeovu transformaciju:

M_2

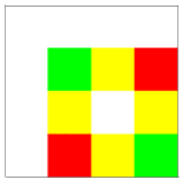
$$\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}$$

M_3

$$\begin{matrix} 1.00 & 1.00 & 1.00 \\ 1.00 & -0.50-0.87i & -0.50+0.87i \\ 1.00 & -0.50+0.87i & -0.50-0.87i \end{matrix}$$

M_4

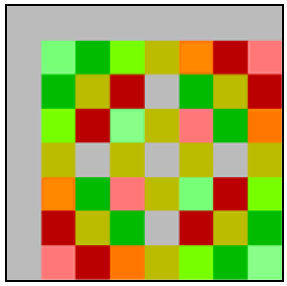
$$\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{matrix}$$



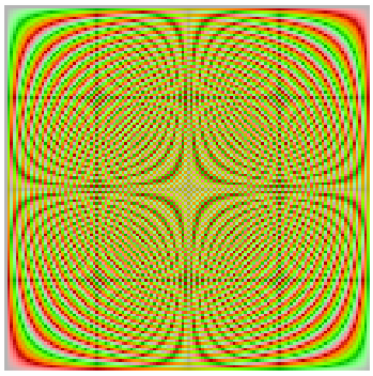
M_8

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \bar{z} & -i & -z & -1 & -\bar{z} & i & z \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & -z & i & \bar{z} & -1 & z & -i & -\bar{z} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\bar{z} & -i & z & -1 & \bar{z} & i & -z \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & z & i & -\bar{z} & -1 & -z & i & \bar{z} \end{matrix}$$

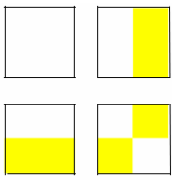
$$\left(z = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \cdot i \right)$$



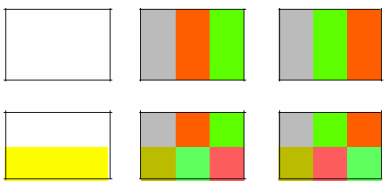
M_{128}



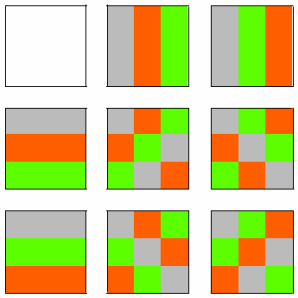
Grafički prikazi baznih slika za Furijeovu transformaciju (slike 3.12, 3.13, 3.14, 3.15, 3.16):



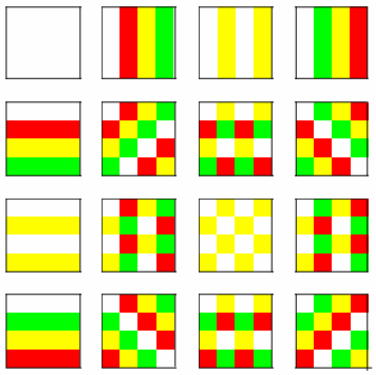
Slika 3.12: Bazne slike za matricu 2×2



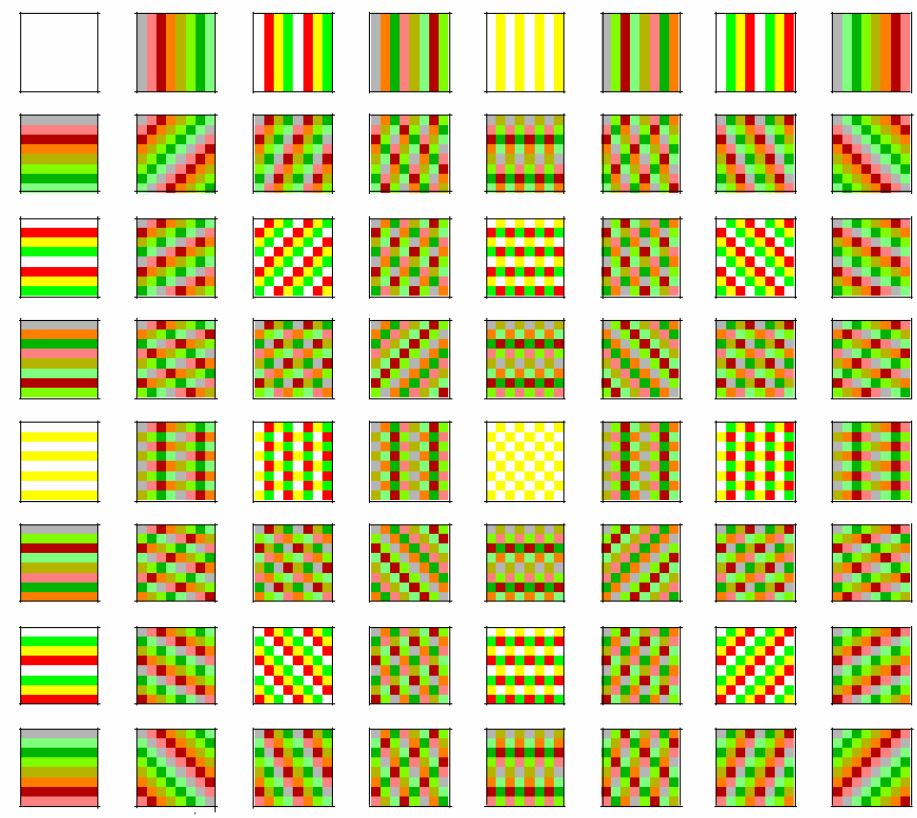
Slika 3.13: Bazne slike za matricu 2×3



Slika 3.14: Bazne slike za matricu 3×3

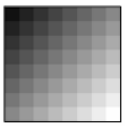


Slika 3.15: Bazne slike za matricu 4×4

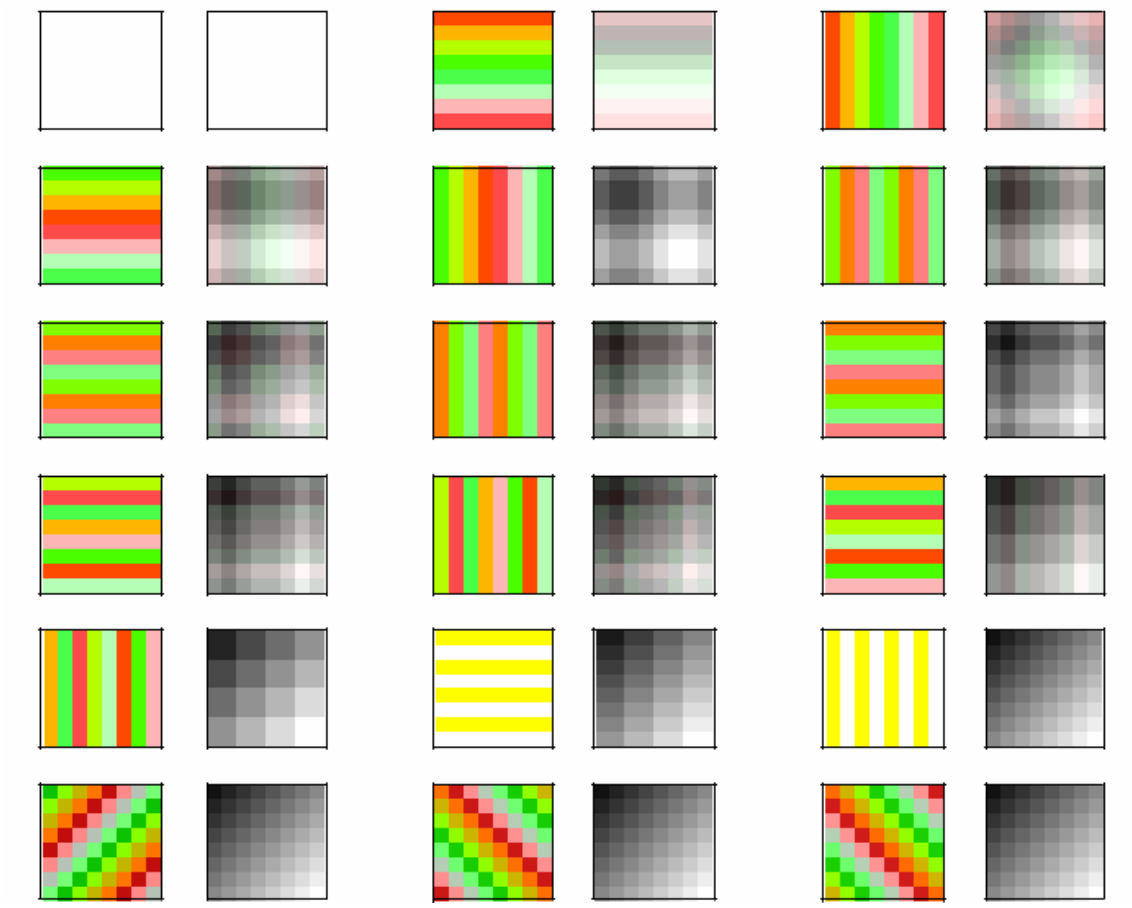


Slika 3.16: Bazne slike za matricu 8×8

Primer 3.5: **Prikaz matrice u novoj bazi.** Na slici 3.17 može se videti nekoliko prvih koraka u predstavljanju slike S pomoću FT baznih slika.

Originalna slika, dimenzije 8×8 piksela 

U ovom primeru matrica S je dimenzije 8×8, i predstavljena je u bazi B_d linearnom kombinacijom 64 bazne slike dimenzije 8×8 (svaka od njih množi se odgovarajućim elementom matrice F). Ovde je prikazano prvih 18 koraka u izvođenju originalne slike iz matrice F i baznih slika. Svaki korak predstavljen je parom (bazna slika, rezultujuća slika). Koraci su uređeni u opadajućem redosledu magnituda koeficijenata u matrici F.



Slika 3.17: Predstavljanje slike pomoću baznih slika u domenu Furijeove transformacije

Furijeova transformacija razlaže sliku u sumu sinusoidnih varijacija sjajnosti, sa svakom mogućom frekvencijom i orijentacijom, i uređuje njihove magnitude i faze tako da odgovaraju originalnoj slici.

Sinusi i kosinusi, koji se pojavljuju u Furijeovom razvoju, periodične su funkcije koje osciluju neograničeno, tj. nisu lokalizovani u prostoru. Zato, Furijeova transformacija signala koji ima nagle promene, generalno zahteva vrlo širok spektar frekvencija[15].

Ako se koristi kompletan skup funkcija koje se dobijaju Furijeovim razvojem, moguće je kompletno rekonstruisati originalnu sliku (korišćenje transformacije samo po sebi ne stvara gubitke). Ali, ako se koristi kompletan skup funkcija, brojevi potrebni da bismo to postigli zauzimaće isto onako mnogo memorijskog prostora kao originalna slika. Ustvari, obično će uzimati i više, jer se u takvom predstavljanju koriste realni brojevi[16], a originalne vrednosti piksela su celobrojne, pa zahtevaju manje memorijskog prostora.

Prednost transformisanja slike u drugi prostor je u činjenici da transformacija grupiše informaciju na drugi način. Furijeov metod razlaže prisutnu informaciju u skladu sa frekvencijom i orijentacijom. Ako vrlo male varijacije frekvencije, kao osenčena mesta u slici, nisu primetne za ljudsko oko, one u transformaciji mogu biti odbačene, prostim anuliranjem vrednosti koje im u domenu transformacije odgovaraju.

DFT je posebno interesantna kada je $m = n$, a n je stepen broja 2. U tom specijalnom slučaju postoji efikasan metod – primena tzv. *brze Furijeove transformacije* (Fast Fourier transform, FFT) [3_08]. Direktno računanje $F(u,v)$ troši $O(n^2)$ aritmetičkih operacija. Broj operacija kod FFT smanjuje se na $O(n \log n)$.

Daleko najpoznatiji FFT je Cooley–Tukey algoritam (J. W. Cooley, J. W. Tukey, 1965) [3_09]. Postoje i drugi FFT algoritmi (FFT algoritam prostog faktora, Bruun-ov, Rader-ov, Bluestein-ov FFT algoritam).

3.2.3. Veza DCT i DFT

Diskretna kosinusna transformacija (DCT) slična je diskretnoj Furijeovoj transformaciji (DFT), ali koristi samo realne brojeve. Ona je ekvivalentna sa DFT približno dvostruke dužine, koja operiše na realnim podacima sa parnom simetrijom (jer je Furijeova transformacija realne i parne funkcije realna i parna), gde su u nekim varijantama ulazni i/ili izlazni podaci šiftovani za pola uzorka.

Mada direktna primena ovih formula iziskuje $O(n^2)$ operacija, moguće je isto izračunati sa samo $O(n \log n)$ faktorizacijom računanja (slično kao kod FFT).

---

[15] Ova primedba važi i za diskretnu kosinusnu transformaciju.

[16] Kompleksan broj se predstavlja parom realnih brojeva.

Ipak, najefikasniji algoritmi računanja DCT, u principu, obično su oni koji su specijalizovani direktno za DCT, nasuprot korišćenju uobičajenog FFT plus $O(n)$ ekstra operacija. Međutim, čak i "specijalizovani" DCT algoritmi su u bliskoj vezi sa FFT algoritmima, **jer je DCT u suštini DFT za realne i parne podatke**, i moguće je napraviti brzi DCT algoritam polazeći od FFT i odbacujući redundantne operacije koje postoje zbog te simetrije.

3.3. Energija slike

Energiju slike S definišemo kao sumu kvadrata vrednosti njenih piksela:

$$E(S) = \sum_{i=1}^{m}\sum_{j=1}^{n} S_{i,j}{}^2 \tag{3.9}$$

gde je slika S zadata matricom piksela $(S_{i,j}, i = 1,...,m;\ j = 1,...,n)$

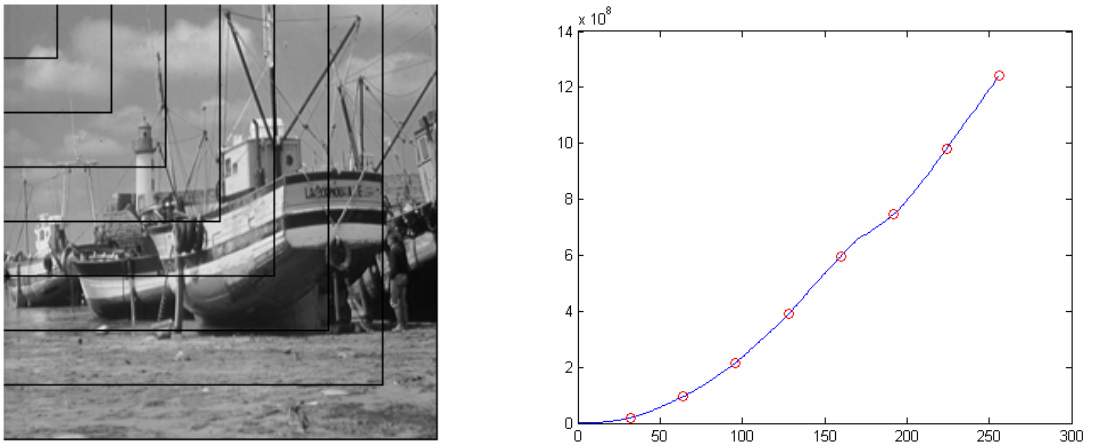
Transformacije kao što je DCT ili DFT (pa i veći broj transformacija talasićima) su unitarne i, prema Parsevalovoj teoremi, za njih važi zakon o očuvanju energije.

Za potrebe ovog teksta, *Parsevalova teorema* se koristi u segmentu vezanom za diskretni signal i transformaciju tog signala. Tako, za unitarnu transformaciju (kao što su diskretna kosinusna i Furijeova), *važi zakon održanja energije*, tj. suma (ili integral) kvadrata funkcije jednak je sumi (ili integralu) kvadrata transformacije. Za sliku dimenzije $m \times n$, u prostornom domenu (S) i u domenu transformacije (T), važi:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} S_{ij}{}^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}{}^2 \tag{3.10}$$

U prostornom domenu, energija je manja u tamnijim, a veća u svetlijim delovima slike, ali ipak te razlike u različitim oblastima slike nisu dramatično velike (slika 3.18).

```
%Energija slike
clear,close all;format compact
C0=double(imread('d:\zig\fishingboat_m.tif'));
[m1,n1]=size(C0);C1=C0;
Ener=sum(sum(C0.*C0))
nizi=1:m1;
for i1=1:m1
    C0i=C0(1:i1,1:i1);
    Ener0(i1)=sum(sum(C0i.*C0i));
end
for i1=32:32:m1
    C1(1:i1,i1)=0;C1(i1,1:i1)=0;
    C1(1:i1+1,i1+1)=0;C1(i1+1,1:i1+1)=0;
end
figure('color','w'),imshow(uint8(C0))
figure('color','w'),imshow(uint8(C1))
figure('color','w'),plot(nizi,Ener0)
i1=32:32:m1;
hold on, plot(i1,Ener0(i1),'or')
```



Slika 3.18: Slika 'Fishingboat', veličine 256x256 piksela, u prostornom domenu, i grafik porasta energije. Apscise predstavljaju dimenziju dela slike za koji se energija računa (kvadrata u gornjem levom uglu) (grafik desno predstavlja prikaz rasta energije slike sa porastom površine slike).

Crne linije na slici nisu sastavni deo same slike, nego označavaju kvadrat – deo slike u kome je vrednost energije prikazana kružićem na grafiku sa desne strane. Ovo je samo ilustracija jasnoće radi – u programu je vrednost energije ustvari računata za sve kvadratne podslike koje sadrže gornji levi piksel slike.

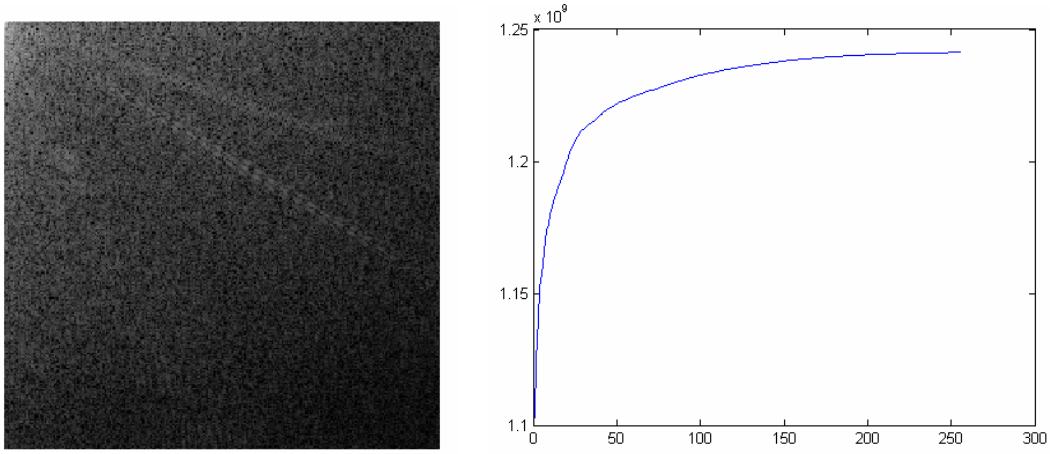
Diskretna kosinusna transformacija čuva ukupnu energiju; energija slike u DCT domenu jednaka je energiji originalne slike. Međutim, energija slike u DCT domenu grupisana je u gornjem levom uglu (gornji levi ugao je mnogo "svetliji" od ostatka DCT slike).

Da bi se izbegla potpuno crna slika (zbog ogromne razlike između vrednosti u okolini DC elementa i ostatka matrice), apsolutne vrednosti elemenata matrice slike su pre prikaza logaritmovane.

```
%boje_log, funkcija za prikaz matrice sa logaritmovanim magnitudama
function boje_log(mat,ind1,ind2,ind3)
matap=abs(mat)+0.0001;
mat1=mat./matap;
mat=mat1.*log2(matap+1);
boje(mat,ind1,ind2,ind3)
```

Za prikaz slike 3.19 korišćen je kod:

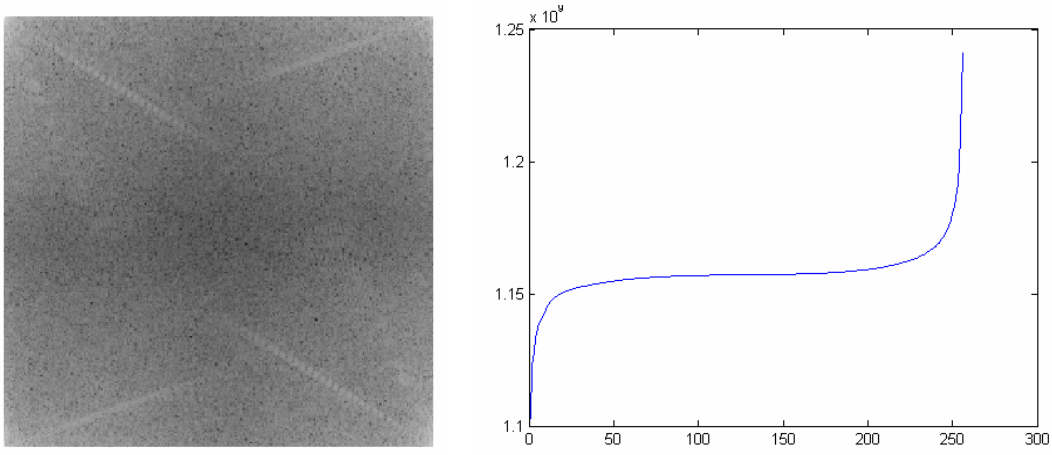
```
DC0=dct2(C0);
Ener1=sum(sum(DC0.*DC0))
nizi=1:m1;
for i1=1:m1
    DC0i=DC0(1:i1,1:i1);
    Ener2(i1)=sum(sum(DC0i.*DC0i));
end
figure('color','w'),boje_log(abs(DC0),1,1,1)
figure('color','w'),plot(nizi,Ener2)
```



Slika 3.19: Logaritmovan spektar snage 17 matrice slike u DCT domenu, i grafik porasta energije. Treba primetiti da grafik ne sadrži koordinatni početak, nego da mu se ordinate kreću od 1.1 do 1.25 ($\times 10^9$).

Za Furijeovu transformaciju, prikaz porasta energije (slika 3.20) predstavljen je sa:

```
FC0=fft2(C0);
Ener3=sum(sum(abs(FC0).^2))/(m1*n1)
nizi=1:m1;
for i1=1:m1
    FC0i=FC0(1:i1,1:i1);
    Ener4(i1)=sum(sum(abs(FC0i).^2))/(m1*n1);
end
figure('color','w'),boje_log(abs(FC0),1,1,1)
figure('color','w'),plot(nizi,Ener4)
```

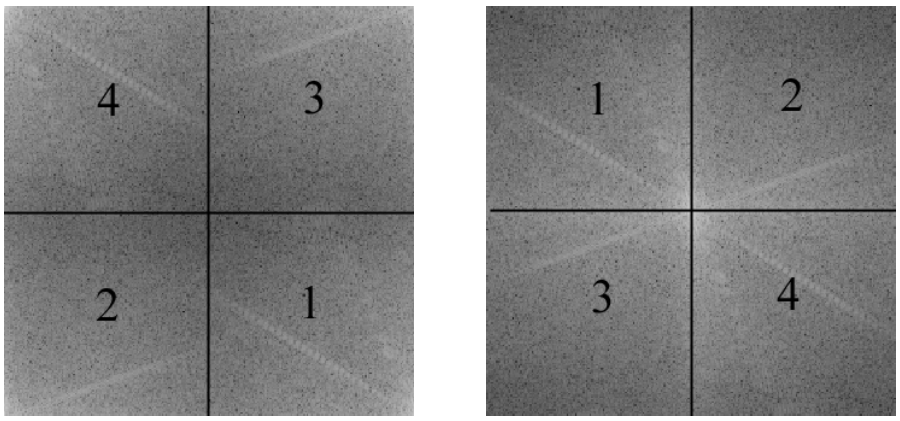


Slika 3.20: Logaritmovan spektar snage matrice slike u domenu Furijeove transformacije, i grafik porasta energije. Treba primetiti da ni ovaj grafik ne sadrži koordinatni početak.

---

[17] *Spektar snage* (engl. *power spectrum*) matrice predstavlja sliku apsolutnih vrednosti njenih elemenata, tj. ako je matrica $M = (M_{ij})_{i=1:m, j=1:n}$, onda je spektar snage jednak $M_{PS} = (|M_{ij}|)_{i=1:m, j=1:n}$. Kod matrica nastalih diskretnom kosinusnom (odnosno Furijeovom) transformacijom, raspon vrednosti elemenata u matrici M_{PS} je vrlo velik; prikaz sa leve strane ustvari predstavlja matricu $[\log_2(1+|M_{ij}|)]_{i=1:m, j=1:n}$

Grafici funkcija pokazuju koliko je DC koeficijent (gornji levi element) u domenu transformacije veći od ostalih. DC koeficijenat ustvari predstavlja sumu svih vrednosti matrice slike u prostornom domenu, podeljenu korenom broja piksela na slici. Ukupna energija je u sva tri ovde prikazana domena, ista (1.2413e+009). Međutim, dok je u prostornom domenu manje–više ravnomerno raspoređena u celoj slici, u slučaju transformacije ona je skoncentrisana na jednom mestu, i to u okolini DC koeficijenta.
Dok rečeno, dosta očigledno važi za DCT, kod Furijeove transformacije je potrebno dodatno objašnjenje. U Furijeovom domenu energija je sakupljena uglavnom u uglovima (za razliku od DCT, u sva četiri ugla). Međutim, DFT (kao uostalom i DCT) pretpostavlja periodičnost. To su transformacije koje pretpostavljaju da se originalna slika periodično ponavlja (ustvari, "popločava" ravan). Isto važi i za slike u domenu transformacije. Zato se ništa ne gubi (a u nekim slučajevima je korisno) ako se slika u Furijeovom domenu "centrira", tj. preuredi tako da se uglovi nađu u centru (slika 3.21).



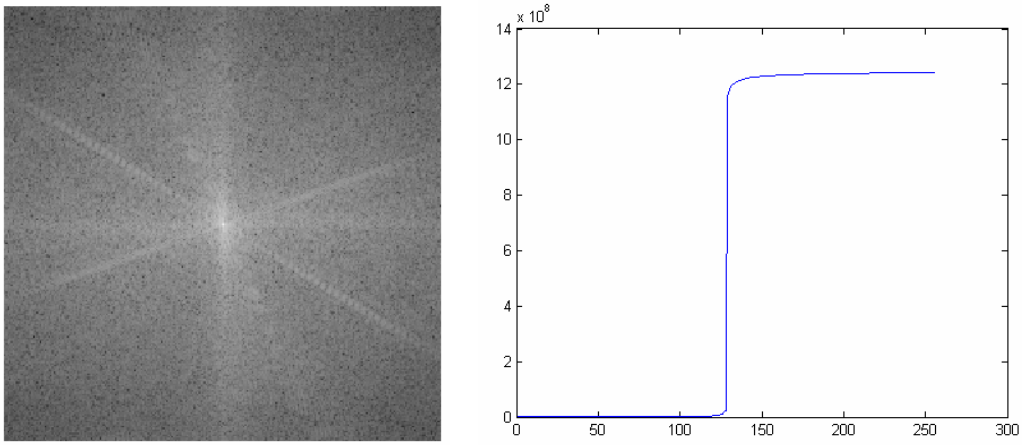
Slika 3.21: Ilustracija centriranja; centrirana slika u domenu Furijeove transformacije

Funkcija *cen* koja obavlja ovo centriranje data je kodom:

```
%cen: centriranje matrice Furijeove
%transformacije;
function FCC=cen(FC0);
[m1 n1]=size(FC0);
m2=m1/2;n2=n1/2;
FCC(1:m2,1:n2)=FC0(m2+1:m1,n2+1:n1);
FCC(m2+1:m1,n2+1:n1)=FC0(1:m2,1:n2);
FCC(1:m2,n2+1:n1)=FC0(m2+1:m1,1:n2);
FCC(m2+1:m1,1:n2)=FC0(1:m2,n2+1:n1);
```

Dodatak prethodnom programu – prikaz "centrirane" DFT i porasta njene energije (slika 3.22):

```
CFC0=cen(FC0);
for i1=1:m1
    CFC0i=CFC0(1:i1,1:i1);
    Ener5(i1)=sum(sum(abs(CFC0i).^2))/(m1*n1);
end
figure('color','w'),boje_log(abs(CFC0),1,1,1)
figure('color','w'),plot(nizi,Ener5)
```



Slika 3.22: Centrirana slika u Furijeovom domenu, i njen grafik raspodele energije

Kod slike 'Fishingboat', DC koeficijent u DCT domenu iznosi 3.3208e+004, što znači da je njegova energija jednaka čak 1.1027e+009, ili da se 88.84% ukupne energije ove slike nalazi u DC koeficijentu.

Vrednosti koeficijenata Furijeove transformacije bi, da bi poređenje bilo odgovarajuće, trebalo podeliti sa kvadratnim korenom ukupnog broja piksela. U tom slučaju, DC koeficijent Furijeove transformacije bio bi isti kao kod DCT, i svi navedeni brojevi bi važili i ovde.

88.84% ukupne energije nalazi se u DC koeficijentu slike 'Fishingboat', dimenzije 256×256 piksela. U slučaju prvog 8x8 bloka te slike, u njegovom DC koeficijentu nalazi se čak 99.97% ukupne energije slike bloka!

Tumačenje opravdanosti konverzije slike u domen transformacije pri kompresiji: **Transformacija prevodi matricu slike u oblik u kome je gotovo sva energija slike sakupljena u malom broju susednih koeficijenata. Zatim se iz preostalog dela ove matrice odbacuju (bez velikog uticaja na energiju slike) "manje važni za ljudsko oko" podaci.**

3.4. Kompresija talasićima

Talasić (*wavelet*) je funkcija koja ima srednju vrednost 0, i vrednosti vrlo bliske 0 svuda osim na ograničenom intervalu.

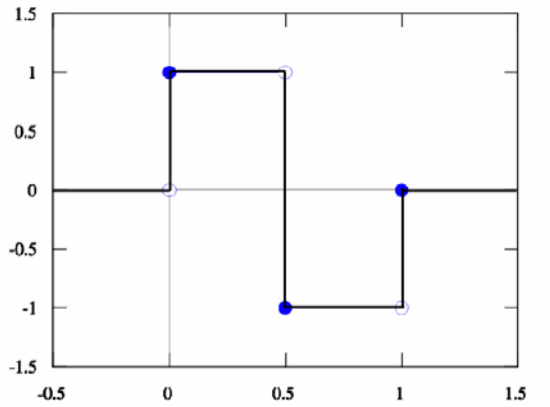
U kompresiji talstićima polazi se od jedne funkcije talasića – *matičnog (mother) talasića*. Matični talasić se po potrebi skalira i translira da bi dao nove talasiće – učesnike u aproksimaciji date funkcije [3_10, 3_11, 3_12].

Za razliku od DCT i DFT, funkcija na kojoj talasić počiva nije periodična nego je lokalizovana u prostoru (izvan određenog intervala jednaka je 0).

Kao rezultat, talasići imaju prednosti nad tradicionalnim Furijeovim metodama u analiziranju fizičkih situacija u kojima signal sadrži prekide i nagle promene. Signali sa naglim promenama mogu biti rekonstruisani sa mnogo manjim skupom baznih funkcija talasića nego u slučaju u kom se predstavljaju periodičnim funkcijama. Zato su oni ekonomičniji od diskretne kosinusne ili Furijeove transformacije.
Bazne funkcije talasića obično su takve da omogućuju lako odvajanje glatkih komponenti od komponenti detalja. To može da pomogne u kompresiji slika, gde je komponente detalja moguće odbacti bez ozbiljne degradacije slike.

Harov talasić (slika 3.23) je najjednostavniji moguć talasić. Uveo ga je Alfred Haar 1909. Sam termin "wavelet" skovan je znatno kasnije[18]. Osnovna mana Harovog talasića je da on nije neprekidan, pa zato ni diferencijabilan. Možemo ga opisati kao skok funkciju $f(x)$ sa

$$f(x) = \begin{cases} 1 & 0 \le x < 1/2, \\ -1 & 1/2 \le x < 1, \\ 0 & \text{izvan ovih intervala} \end{cases}$$



Slika 3.23. Harov talasić

Sledi ilustracija razlaganja jednodimenzionog niza uz pomoć Harovog talasića.
Neka se dati diskretni signal sastoji od 4 vrednosti:

$$(y_1, y_2, y_3, y_4) = (5,-1,2,0)$$

Ispituju se vrednosti signala u parovima, i razlažu u prosek i ostatak:

$$(5,-1) = (2,2) + (3,-3) \qquad (2,0) = (1,1) + (1,-1)$$

Prosek treba posmatrati kao glatku komponentu para, a *ostatak* – kao detalj. Zatim se ispituje glatki deo parova i razlaže na sličan način:

[18] Uveo ga je Jean Morlet, sredinom sedamdesetih godina prošlog veka.

$$(2,2,1,1) = (1.5,1.5,1.5,1.5) + (0.5,0.5,-0.5,-0.5)$$

Prvi vektor je najglatkiji deo signala. Rezultat grupisanja svih komponenti daje:

$$(5,-1,2,0) = 1.5(1,1,1,1) + 0.5(1,1,-1,-1) + 3(1,-1,0,0) + 1(0,0,1,-1)$$

Ovde je prikazan signal kao linearna kombinacija baznih funkcija talasića. Rezultujuća transformacija talasićima može se zapisati kao vektor četiri amplitude:

$$\vec{Y} = (S, D, d_1, d_2) = (1.5,0.5,3,1)$$

gde S označava najglatkiji deo, D detalj prvog nivoa, a d_1, d_2 detalje drugog nivoa.

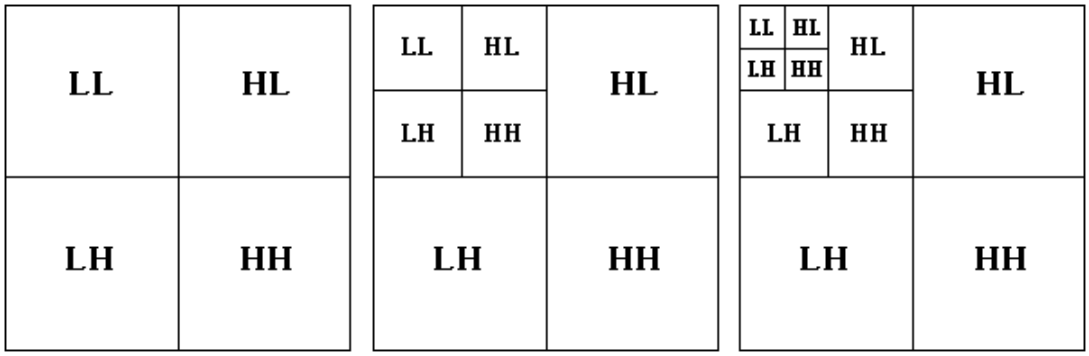
Kod dužeg signala (od n vrednosti – n je celobrojni stepen broja 2), nastavlja se hijerarhija. Polazi se od parova uzastopnih vrednosti da bi se dobio, kao i u gornjem primeru, glatki signal prvog nivoa od $n/2$ vrednosti, pa glatki signal drugog nivoa od $n/4$ vrednosti, itd. Postupak se nastavlja dok se ne dođe do glatkog signala najvišeg nivoa koji je baš prosek. Komponente detalji na svakom koraku čuvaju se kao konačni rezultat.

Inverzna transformacija može biti lako konstruisana obrćući korake uključene u direktnu transformaciju.

Isti algoritam može se primeniti i u dvodimenzionom slučaju, na slike. Sledi prikaz koraka primene Harovih talasića na redove, pa na kolone slike. Inače, procedura diskretne transformacije talasićima je sledeća: Odaberu se niskopropusni[19] i visokopropusni filter, takvi da među sobom tačno polove oblast frekvencije. Primenom ova dva filtra na svaki red podataka, dobijaju se podaci u redu takvi da prva polovina reda sadrži niske, a druga – visoke frekvence. Procedura se obavlja za sve redove.

Zatim se obavlja filtriranje za svaku kolonu međupodataka. Rezultujuća matrica koeficijenata sadrži 4 opsega (bands) podataka, označena sa LL (low–low), HL (high–low), LH i HH. LL podaci se dalje mogu razložiti na isti način, čime se dobija još podopsega podataka. Može se nastaviti i dalje, i time se dobija rezultujuće piramidalno razlaganje kao na slici.

[19] *Niskopropusni filter* propušta dosta dobro niske frekvencije, a oslabljuje ili blokira visoke frekvencije. *Visokopropusni filter* propušta visoke, a blokira niske frekvencije. (Šta su visoke, a šta niske frekvencije, zavisi od trenutne situacije – nema određene granice).



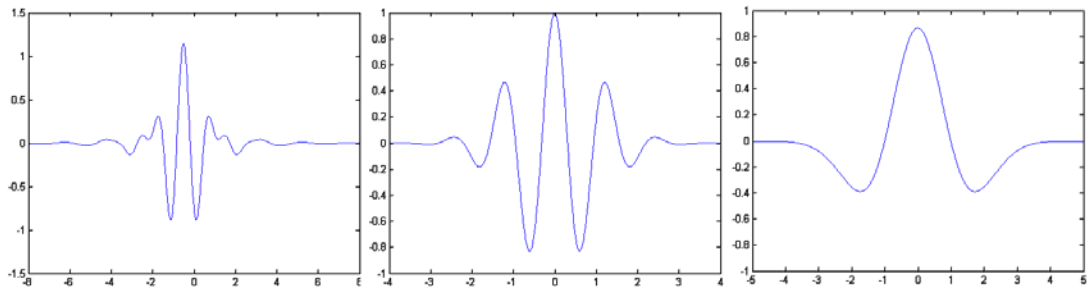
Slika 3.24: Piramidalno razlaganje talasićima: razlaganje jednog nivoa; razlaganje dva nivoa; razlaganje tri nivoa

LL opseg podataka sadrži *aproksimaciju*, a svi ostali opsezi sadrže *detalje*.

Rezultati ovakvog razlaganja predstavljeni su slikama 3.24 i 3.25.

Slika 3.25: Ilustracija procesa razlaganja talasićima (dva nivoa) za sliku 'Cameraman'.

Harovi talasići su grubi izlomljeni talasi. Sofisticiraniji talasići su glatkiji. Neki su prikazani na slici 3.26.



Slika 3.26: Primeri talasića: *Meyer*, *Morlet*, *Mexican hat*

Proces kompresije slika talasićima čine tri koraka:

- razlaganje podataka slike – diskretna transformacija talasićima (DWT),

- kvantizacija transformisanih koeficijenata i
- kodiranje kvantizovanih transformisanih koeficijenata.

Razlaganje slike je proces bez gubitaka, koji konvertuje podatke slike iz prostornog u domen frekvencije, gde su transformisani koeficijenti dekorelirani. Gubitak informacije nastaje u koraku kvantizacije, a kompresija se postiže u koraku kodiranja.

Tokom kvantizacije, svaki opseg se kvantizuje različito, zavisno od njegove važnosti. Da bi se postigla zadata mera kompresije, koristiće se grubi kvantizatori (veliki koraci kvantizacije) da bi se kvantizovali opsezi niske energije; koristiće se finiji kvantizatori (manji koraci kvantizacije) da bi se kvantizovali opsezi visoke energije.

Talasići su u osnovi bolji od JPEG-a. Odlični su za slike, muziku i video. Svi proizvodi kompresije talasićima su za sada vlasnički, i postoji mnogo verzija koje dolaze od komercijalnih i akademskih istraživanja po svetu.

Osnovne prednosti talasića nad JPEG kompresijom su:

- Kompresija talasićima, kao novija, primenjuje se na celu sliku, a ne blokove, pa se pri intenzivnoj kompresiji na slici ne vide artifakti blokiranja.
- Transformacija talasićima, za razliku od DCT (i DFT), ne podrazumeva periodičnost. Funkcije korišćene u transformaciji talasićima su lokalizovane u prostoru, i mogu se po potrebi skalirati (umanjivati). Na taj način sa mnogo manje utrošenih podataka, moguće je postići isti efekat kao kod DCT i DFT. To za posledicu ima i manju veličinu komprimovanog fajla.
- DWT se može obaviti u samo $O(n)$ operacija

Zbog očiglednih prednosti u odnosu na JPEG, noviji grafički fajl formati uglavnom u kompresiji koriste talasiće. To važi za formate kao što su PDF i DjVu, ali i za standard JPEG 2000, koji umesto DCT preporučuje talasiće u kompresiji slika.

I pored svih prednosti talasića u odnosu na JPEG, treba imati na umu da je JPEG afirmisan standard sa mnogo raspoloživog, pa i besplatnog softvera, pa se i dalje široko koristi. **Takođe, mnogi algoritmi kompresije talasićima su patentirani, što otežava, pa čak i onemogućuje da se slobodno koriste u drugim softverskim projektima.**

JPEG 2000

JPEG 2000 je standard kompresije slika, zasnovan na talasićima. Nastao je u 2000, sa ciljem da zameni JPEG standard iz 1992. (zasnovan na DCT). JPEG 2000 publikovan je kao ISO standard, ISO/IEC 15444 ([3_13]).

Slika se (ako je u boji) prvo razlaže u tri slike u nijansama sive, koje se zasebno dalje komprimuju.

Svaka slika se zatim deli u tzv. *pločice (tiles),* pravougaone oblasti koje se zatim transformišu i kodiraju zasebno. Pločice mogu biti proizvoljne veličine, a moguće je da se cela slika tretira kao jedinstvena pločica. Sve pločice slike imaju istu veličinu (osim možda onih na desnoj i donjoj ivici slike). Razlog ovakve deobe je manja memorija neophodna za dekodiranje slike. Mana takvog pristupa je umanjeni kvalitet i bloking artifakti (slični kao u starom JPEG standardu).

Na pločice se zatim primenjuje transformacija talasićima do proizvoljne dubine. JPEG 2000 koristi dve različite transformacije talasićima:

- *ireverzibilna* (kompresija sa gubicima): ***CDF 9/7*** transformacija (uvodi kvantizacioni šum).
- *reverzibilna* (bez gubitaka): verzija biortogonalne ***CDF 5/3*** transformacije, koja koristi samo celobrojne koeficijente, pa kvantizacija nije potrebna (odnosno, korak kvantizacije je 1).

Posle kvantizacije obavlja se kodiranje.

JPEG 2000 ni danas nije široko podržan u Web pregledačima, pa se i ne koristi široko na Web-u.

DjVu

DjVu (*déjà vu*) je format, predviđen pre svega za čuvanje skeniranih dokumenata, posebno onih koji sadrže kombinaciju teksta, crteža i fotografija. To omogućuje da se čitljive slike visokog kvaliteta čuvaju u minimalnom prostoru (to je dobro svojstvo i za korišćenje na Webu) [3_14, 3_15].

DjVu nije u osnovi slikovni format, mada se koristi za stvaranje komprimovanih strana knjige. Ne koristi se za slike u HTML dokumentima. On je najbolji izbor za čuvanje elektronskih knjiga, nastalih skeniranjem (bilo crno–belih, bilo u punoj boji). Te skenirane strane komprimuje da stvori vrlo snažno komprimovane fajlove. Omogućuje i čuvanje OCR rezultata uz slike stranica, što čini da se tekst u knjizi može i pretraživati.

Nastao je kao alternativa PDF-u. Za većinu skeniranih dokumenata stvara manje fajlove od PDF-a.

DjVu tehnologiju originalno su razvili Yann Le Cun, Léon Bottou, Patrick Haffner i Paul G. Howard u AT&T Laboratories, 1996. DjVu je slobodan format.

DjVu deli sliku na nekoliko različitih slika, i komprimuje ih posebno. Da bi se stvorio DjVu fajl, početna slika se razlaže u tri: pozadinu, prednji plan i masku. *Pozadina* i *prednji plan* su slike male rezolucije (na pr. 100 dpi). Obično se komprimuju koristeći algoritam kompresije talasićima ***IW44***. Slika *maske* je binarna, visoke rezolucije (na pr. 300 dpi) i sadrži tekst. Maska se komprimuje korišćenjem metoda JB2 (slično sa JBIG2[20]). Metod kodiranja JB2 identifikuje približno identične oblike na strani, kao što su višestruka pojavljivanja određenog znaka u datom fontu, stilu i veličini. On komprimuje bitmapu svakog oblika posebno, i zatim kodira lokacije gde se svaki oblik pojavljuje na strani. Tako, umesto da se slovo "e" komprimuje više puta, komprimuje se samo jednom, i pamti se svako mesto na stranici gde se ono pojavljuje.
Godine 2002, DjVu fajl format je izabran u Internet Archive[21] kao format u kome će njen *Million Book Project* omogućiti korišćenje knjiga na mreži.

DjVu Solo 3.1 je slobodni softver za Windows koji omogućuje korišćenje DjVu formata. Stvoren je 2001. u softverskoj kompaniji LizardTech. On za različite fajlove predviđa različite vrste kompresija:

- Vrlo mala veličina fajlova postiže se za kolor dokumente pomoću kompresija *Scanned* i *Clean*. Dokument se deli u prednji plan (tekst i oštre linije) i pozadinu (boja pozadine i boje, crteži bez oštrih linija).
- *Bitonal* se koristi za crno–bele slike; mnogo bolje prikazuje crteže sa nepravilnim ivicama nego što to čine *Scanned* i *Clean*.
- *Photo* kompresija stvara znatno veće fajlove (približno kao pola veličine *jpg* slike), ali je kvalitet slike uporediv sa originalnim skenom. Ova kompresija je pogodna za fotografije i slike sa mnogo detalja. Kodira se samo pozadina, ali sa visokom rezolucijom. Kolor mape i crteži se dobro komprimuju korišćenjem *Scanned* ili *Clean***.**

3.5. Kompresija fraktalima

Fraktal je geometrijski objekat koji može biti izdeljen u delove, od kojih je svaki sličan originalnom objektu. Ponavlja se na svakoj skali i zato ne može biti prikazan klasičnom geometrijom. Kaže se i da je fraktal *samosličan*, jer je tačno ili približno sličan delu

[20] *JBIG2* je kompresioni standard za binarne slike, razvijen u Joint Bi-level Image Experts Group.

[21] *Internet Archive (IA)* je neprofitna organizacija koja se bavi stvaranjem i održavanjem slobodno dostupne digitalne biblioteke, uključujući i arhivu za World Wide Web.

sebe samog. U mnogim slučajevima fraktal se može generisati ponavljajućim obrascem, tipično rekurzivnim ili iterativnim procesom.

Teorija fraktala ima dugu istoriju. Tehnike zasnovane na fraktalima prvobitno su korišćene u nekoliko oblasti obrade digitalnih slika kao što su segmentacija slika, sinteza slike i računarska grafika, a tek kasnije primena se proširila na kompresiju. Fraktal je geometrijska forma sa jedinstvenim svojstvom da ima ekstremno visoke samostalne nepravilne detalje, uz sadržaj vrlo malo informacija. Na bazi različitih karakteristika fraktala razvijeno je nekoliko metoda kompresije slika [3_16, 3_17].

Kompresija fraktalima zasniva se na činjenici da su neki delovi na slici među sobom slični. Omogućuje kompresiju za slike u boji od 100:1 i veću. Posebno je dobar za fotografije prirodnih scena (drveće, oblaci, reke).

Do pre nekoliko godina, izgledalo je da su fraktali jedna od onih tehnologija sa "velikom budućnošću ispred nje". To je bila obećavajuća tehnika kasnih 1980-tih, kada je pod nekim okolnostima komprimovala mnogo bolje od JPEG-a, svog tada glavnog konkurenta.

Međutim, kompresija fraktalima nikada nije dostigla masovnu upotrebu. Verovatni problem je njena pokrivenost patentima (JPEG se mogao koristiti bez ikakvih patentnih ograničenja). Osim toga, kompresija fraktalima je znatno sporija od JPEG kompresije (dekompresije se uglavnom obavljaju jednakim brzinama). Takođe, popravljena mera kompresije može biti iluzija. Fraktali imaju veliku prednost nad JPEG-om samo na niskom kvalitetu kompresije, koji je obično nepoželjan. Činjenica da fraktalima komprimovane slike, kada se uvećaju preko njihove originalne veličine, izgledaju bolje od slično uvećanih JPEG slika, predstavlja takođe nebitnu razliku.

Pokazalo se i da je za najimpresivnije primere kompresije fraktalima potrebna značajna ljudska intervencija. Proces generisanja slike iz fraktala je lako automatizovati, ali je obrnuta procedura, generisanje optimalnog fraktalnog prikaza iz slike, visoko netrivijalna. Većina slika realnog sveta ima različita matematička svojstva. Na primer, bilo bi dobro da se planine, oblaci i drveće predstave sa nekoliko različitih klasa fraktala.

Danas kompresija fraktalima izgleda i manje bitna, jer je kompresija talasićima nadmašuje u većini primena (za one koji su voljni da prkose problemima patenata); osim toga, JPEG se i dalje masovno koristi.

***

Osnovni problem u korišćenju kompresije sa gubicima leži u činjenici da se pri naknadnim editovanjima slike, oštećenja akumuliraju. Zato nikada ne treba editovati ovakve slike. Edituje se original, koji se čuva u sirovom (nekomprimovanom) obliku, ili u formatu koji komprimuje bez gubitaka (kao što su TIFF, BMP ili PNG), a kopija sa gubicima se pravi od originala tek u poslednjem trenutku, kada hoćemo da napravimo mali fajl koji ćemo poslati preko mreže.

Slike komprimovane sa gubicima naprosto treba tretirati kao fotokopije, a nekomprimovane (ili komprimovane bez gubitaka) slike, kao originale.

2. deo: Tehnike sakrivanja informacije u digitalne radove

4. Digitalni vodeni žig

Digitalni vodeni žig je obrazac bitova koji se ugrađuje u digitalnu sliku, video ili audio klip, ili neki drugi digitalni rad, a **sadrži neku informaciju u vezi sa tim radom**. Ova informacija obično označava autora ili vlasnika rada. Osnovna namena digitalnog vodenog žiga je **zaštita autorskih prava** nad digitalnim radovima. Praksu sakrivanja informacije o radu u sam taj rad u daljem tekstu će biti nazivana *praksa (digitalnog vodenog) žiga* (engleski termin je *digital watermarking*) [4_01, 4_02, 4_03].

4.1. Malo istorije: Razlozi pojave digitalnog vodenog žiga

Termin *digitalni vodeni žig* vodi poreklo od vodenog žiga na papiru. Prvi takvi žigovi na papiru pojavili su se krajem 13. veka u Italiji; verovatna namena im je bila označavanje proizvođača papira. Termin *vodeni žig* skovan je u 18. veku, u vreme kada se počeo pojavljivati na novčanicama i važnim dokumentima, da bi se sprečilo njihovo falsifikovanje.

Termin *digitalni vodeni žig* pojavio se zajedno sa *praksom žiga*, ranih devedesetih godina prošlog veka. Osnovni razlozi nastanka prakse žiga su

- pojava digitalnih kopija i
- nagli razvoj Interneta.

Dok su muzika i filmovi čuvani na analognim trakama, opasnost za njihove vlasnike nije bila preterano velika. Analogne kopije su uvek bile slabijeg kvaliteta od originala, a kopije druge generacije (kopije kopija) bile su vrlo loše. Digitalne kopije su, međutim, praktično istog kvaliteta kao original. S prelaskom na digitalno čuvanje muzike i filmova, opasnost od piraterije dobila je dramatične razmere.

Internet se mnogo koristi i za trgovinu, a zaštita od krađe ovde je slaba. I još nešto: na Internetu nema ambalaže! Uobičajeno označavanje vlasnika (tekst na kutiji ili nosiocu – gramofonskoj ploči, traci,...) ovde zato ne dolazi u obzir.

Prvi pokušaji da se digitalni sadržaji na Internetu zaštite od krađe zasnivali su se na korišćenju *kriptografije*. Sadržaj se preko mreže slao kriptovan. Samo zakonit kupac

znao je ključ za dekriptovanje. Ako bi neko pokušao da ukrade rad, bez ključa ga ne bi mogao koristiti.

Ovo rešenje ima veliku manu. Podaci su zaštićeni samo dok ne budu dekriptovani. Kada kupac dekriptuje fajl, niko ga ne može sprečiti da ga tako dekriptovanog dalje preprodaje.

Da bi se prava nad intelektualnom svojinom u takvim okolnostima mogla očuvati, intenzivno se traže načini da se u okolnostima nedostatka ambalaže vlasništvo pojedinca nad dokumentom označi, i u slučaju falsifikovanja i neovlašćene distribucije, može braniti na sudu.

Jedno vreme mislilo se da je rešenje *upis u zaglavlje fajla*. Definisao bi se fajl format u kome je žig deo bloka zaglavlja, neuklonjiv bez uništavanja originalnog signala, jer deo definicije fajl formata zahteva da vodeni žig bude unutra. Neki elektronski sistemi za zaštitu autorskih prava predlažu ovakav mehanizam. "Digitalni vodeni žig" koji se pomoću programa Adobe Acrobat može staviti u PDF dokumente je jedno od mesta gde se ovo rešenje koristi. Slabost ovog rešenja je da konverzija u drugi format uklanja žig.

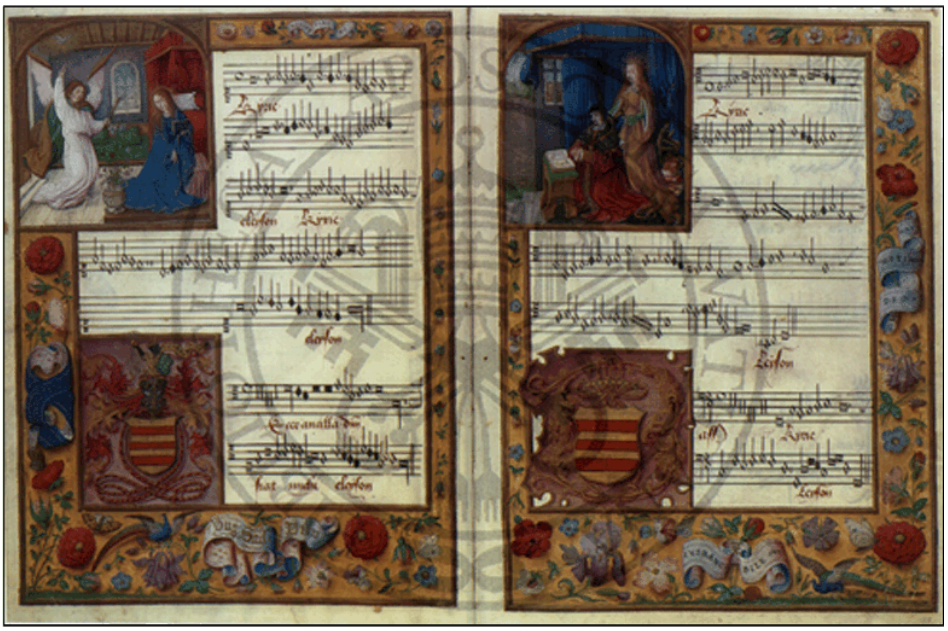
Prirodno se nameće ideja da se informacija o vlasništvu smesti u same podatke digitalnog rada (u slučaju slike, na primer, na ovaj način će se promeniti neki pikseli). Ovo **utiskivanje informacije o vlasništvu u sam signal je** ustvari **praksa** ugradnje **digitalnog vodenog žiga**.

4.2. Primetan i neprimetan žig

Digitalni vodeni žig dokumenata koji se čuvaju u elektronskom obliku pojavljuje se u osnovi u dva vida – kao primetan i kao neprimetan (u slučaju slike vidljiv ili nevidljiv; u slučaju zvučnog fajla, čujan ili nečujan).

Jedan primer primetnog žiga je u video domenu, gde TV mreže stavljaju svoj logo u ugao slike na ekranu.

Sledi primer primetnog žiga za slike. Slika jednog manuskripta iz Vatikanskog muzeja, koja se pojavila na Internetu, ima ugrađen primetni žig, koji podseća na svoje papirnate prethodnike [4_04, 4_05]. Ovde je obaveštenje o vlasništvu očigledno, ali je slika van komercijalnog značaja.



Slika 4.1: Primetni digitalni vodeni žig na manuskriptu iz Vatikanskog muzeja

I primetni i neprimetni žig imaju za cilj zaštitu autorskih prava i vlasništva nad intelektualnom svojinom, ali tu zaštitu obezbeđuju na različite načine. Primetni žig podatak o vlasništvu vrlo uočljivo prikazuje. On, bar u principu, eliminiše komercijalnu vrednost dokumenta za mogućeg kradljivca, bez umanjenja korisnosti dokumenta za zakonite, dozvoljene svrhe.

Neprimetni žig postoji u dokumentu na način koji uopšte ne umanjuje njegov kvalitet. On više pomaže u hvatanju lopova nego u obeshrabrivanju krađe. On povećava verovatnoću uspešnog sudskog gonjenja. Naravno, u izvesnoj meri će odvratiti i potencijalnog kradljivca, ako kod njega postoji svest o mogućem postojanju žiga.

U daljem tekstu će se razmatrati **neprimetan** digitalni vodeni žig.

4.3. Aplikacije digitalnog vodenog žiga

Mogućih primena digitalnog vodenog žiga je mnogo, zahvaljujući njegovim brojnim kvalitetima. On je trajno ugrađen u podatke rada, i nemoguće ga je odstraniti bez velikog narušavanja kvaliteta.

4.3.1. Praćenje emitovanja

Brojne su situacije u kojima postoji interes da se zna da li je određen sadržaj emitovan na televiziji ili nije. Tako, oglašavači su zainteresovani da TV stanice emituju reklame njihovih proizvoda sve vreme koje su oni platili. S druge strane, autori sadržaja koji se emituje na televiziji žele da znaju da li je TV stanica emitovala njihov rad više puta

nego što im je platila, kao i da li neka piratska TV stanica emituje neovlašćeno njihov rad.

Digitalni vodeni žig daleko je lakše čuvati u bazi sadržaja koji bi se u ovu svrhu pratili, nego na primer, ceo film od jednog sata. Mana rešenja ugradnje žiga u podatke digitalnog rada je dodatna cena ove ugradnje.

4.3.2. Identifikacija vlasnika

Ugradnja žiga je način da se, u nedostatku bilo kakve ambalaže, označe vlasnici digitalnih radova na Internetu. Na taj način će savesni korisnik moći da pre korišćenja tuđeg rada, pita vlasnika za dozvolu. Sa ovom primenom u svesti, Digimarc je svoj ugrađivač i detektor žiga uključio u Adobe Photoshop. Tako je moguće označiti vlasništvo nad slikom pre njenog stavljanja na Web sajt. Takođe, lako je saznati ko je vlasnik neke slike nađene na Internetu, da bi se zatražila njegova dozvola za korišćenje (pod uslovom da je u tu sliku ugrađen Digimarc-ov žig).

Treba ipak napomenuti da Digimarc-ov žig može da se koristi samo za obaveštavanje savesnih korisnika ko je vlasnik slike. Njega nije moguće koristiti kao dokaz na sudu, jer nije bezbedan od neprijateljskog falsifikovanja i uklanjanja.

4.3.3. Dokaz vlasništva

Svakako najinteresantnija upotreba digitalnog vodenog žiga je njegovo korišćenje kao dokaz vlasništva u sudskom sporu. On u svakom slučaju, ima kvalitete da se kao takav koristi, baš kao što se otisci prstiju i uzorci krvi koriste kao dokaz u sudu. Naravno, za takvu primenu, on mora posedovati određene osobine, koje ga čine otpornim na različite neprijateljske napade, usmerene na ugrožavanje svrhe žiga.

4.3.4. Dokaz autentičnosti

Za digitalni rad koji treba da se upotrebi kao dokaz na sudu (snimak kamere za nadzor, ili fotografija snimljena na mestu zločina), mora biti sigurno da sadržaj posle snimanja nije menjan. Na primer, nije suviše teško u programu kao što je Photoshop nekoga izbrisati sa fotografije, ili izmeniti broj registarske tablice automobila.

Problem provere autentičnosti ovde se rešava slično kao u kriptografiji, stvaranjem digitalne signature. *Digitalna signatura* je u suštini kriptovani rezime poruke.

Tehnologiju digitalne signature je na digitalne kamere primenio Friedman, koji je sugerisao stvaranje "pouzdane kamere", računanjem signature unutar kamere. Kako

samo kamera ima ključ za stvaranje signature, to će svaka kasnija intervencija na slici učiniti da digitalna signatura više ne odgovara slici.

4.3.5. Kontrola kopiranja

Američki zakon dozvoljava da se TV emisije snime pomoću video rikordera, radi kasnijeg gledanja. Kopije tih kopija, međutim, nisu dozvoljene.

Moguće rešenje je da se video rikorderima dopusti da snimaju samo sadržaj koji ima odgovarajući žig koji to snimanje dozvoljava. Ako se u sadržaj televizijskih emisija ugradi lomljiv žig koji se pri prvom presnimavanju uništava, problem će biti rešen, naravno, ukoliko proizvođači video rikordera budu ovu kontrolu hteli da uključe u svoje proizvode (dodavanjem ovog ograničenja, proizvođači ustvari smanjuju vrednost svojih proizvoda).

4.3.6. Praćenje transakcija

Autor digitalnog rada može, ako želi da u prodaji kopija tog rada koristi usluge specijalizovanih Web sajtova, da pre slanja svog rada na njihove adrese, u svaku kopiju, osim žiga koji označava njega kao nosioca autorskih prava, ugradi poseban žig, koji jedinstveno označava sajt na koji je upućen.

Urednici Web sajtova koji šalju primerke rada kupcima, u ove kopije će za svakog kupca ugraditi drugačiji žig. Ukoliko kasnije neki od kupaca počne da piratizuje fajl, moći će se saznati ko je to uradio.

Ova aplikacija često se tretira posebno od ostalih navedenih aplikacija, zbog većeg broja različitih žigova koji se ugrađuju u jedan rad. Uobičajeni naziv za nju je *fingerprinting*.

4.4. Zahtevi za dobar žig

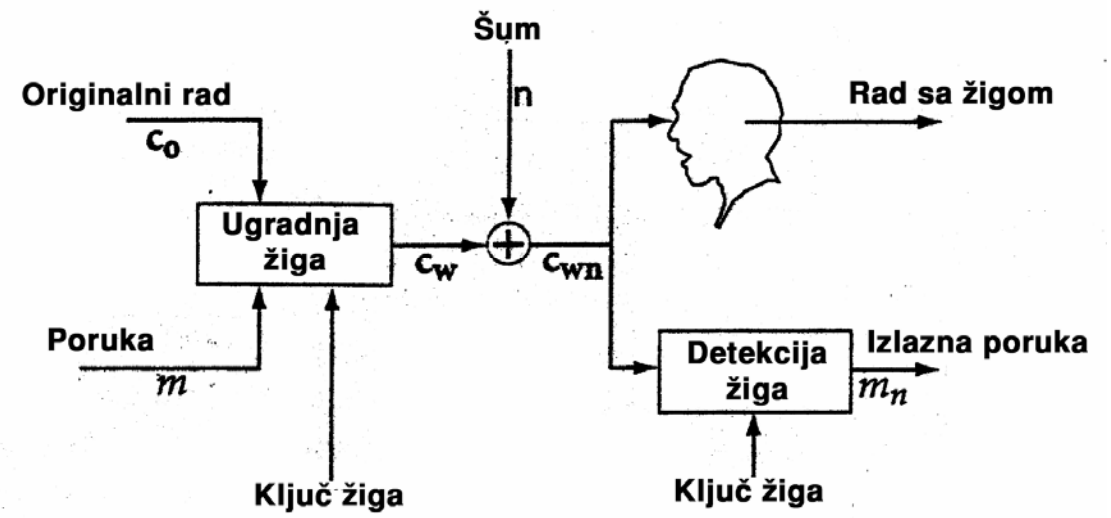
Dobar neprimetni žig nije lako napraviti. On treba da zadovolji određene, međusobno u velikoj meri suprotstavljene uslove:

- **Vernost**: Promene u digitalnom radu (zbog ugradnje žiga) treba da budu neprimetne za slučajnog posmatrača.
- **Robusnost**: Žig treba da je otporan na uobičajene operacije nad digitalnim radom. Šta su uobičajene operacije, zavisi od toga šta je rad. U slučaju slike to mogu biti promena sjajnosti i kontrasta, kompresija sa gubicima, skaliranje, rotacija, opsecanje,...

- **Sigurnost**: Žig treba da bude otporan na pokušaje neprijatelja da ometu njegovu svrhu. Mogući napadi su neovlašćena detekcija, neovlašćena ugradnja i neovlašćeno uklanjanje.

4.5. Praksa žiga kao komunikacija

Praksu digitalnog vodenog žiga moguće je shvatiti kao komunikaciju. Na početnom i završnom kraju komunikacionog kanala su *ugradnja* i *detekcija* žiga. Svi događaji vezani za rad sa žigom od momenta ugradnje, pa do detekcije, dešavaju se u komunikacionom kanalu. Izobličenja koja u kanalu digitalni rad pretrpi (uobičajene operacije, ali i neprijateljski pokušaji da se žig ukloni) obično se modeliraju kao *šum*.



Slika 4.2: Praksa žiga kao komunikacija

Na jednom kraju komunikacionog kanala nalazi se *ugrađivač*. Ulazni podaci za ugrađivač su originalni rad c_0 i poruka žiga m (i možda još neki podatak). Izlaz iz ugrađivača je rad sa žigom c_w.

Na drugom kraju komunikacionog kanala nalaze se dva vrlo različita posmatrača: *čovek* i *detektor žiga*. Čovek svojim čulima prima rad sa žigom c_{wn} (možda deformisan nekim šumom u komunikacionom kanalu). To što opaža doživljava kao nešto što liči na originalni rad.

Detektor žiga kao ulaz prima (možda deformisan) rad sa žigom c_{wn} i možda još neki podatak, a kao izlaz izdaje detektovanu *poruku žiga* m_n. Ukoliko je žig bio robustan na šum kome je rad bio izložen u komunikacionom kanalu, biće $m_n = m$.

U zavisnosti od toga da li se (i kako) u ugradnji i detekciji koristi originalni rad, ugrađivač i detektor mogu biti *slepi* ili *informisani*.

Ako se u algoritmu ugradnje (radi poboljšanja njegove efikasnosti) koristi znanje o osobinama originalnog rada (u koji se žig ugrađuje), *ugradnja* je *informisana* – u suprotnom je *slepa*.

Informisani detektor kao ulazni podatak (osim rada sa žigom) koristi i originalni rad. Informaciju o ugrađenom žigu moguće je lako dobiti kombinovanjem rada bez žiga i rada sa žigom. *Slepi detektor* ne koristi informaciju o originalnom radu. U velikom broju aplikacija detektori su slepi, jer najčešće u fazi detekcije originalni rad nije dostupan.

Ugrađivač i detektor prikazani na prethodnoj slici su slepi, jer ne koriste na opisane načine originalni rad. Prikazano je rešenje u kome ugrađivač i detektor kao dodatni podatak koriste ključ žiga. *Ključ žiga* je tajni broj, poznat samo ovlašćenom licu. Služi kao seme generatora niza pseudoslučajnih brojeva, koji se na neki način koriste u algoritmima ugradnje i detekcije.

5. Praksa žiga, steganografija, kriptografija

U sakrivanju informacija u digitalne radove i tajnoj komunikaciji, našle su svoje mesto tri discipline – kriptografija, praksa žiga i steganografija.

5.1. Steganografija, kriptografija i praksa žiga

Kriptografija je disciplina koja se bavi metodama sakrivanja **sadržaja** informacije od neovlašćenih lica.

Cilj *steganografije* je sakrivanje tajne poruke, tako da i samo njeno **postojanje** ostaje tajna.

Praksa žiga se bavi ugradnjom poruke o **autorskim pravima** nad digitalnim radom u sam taj rad, na način koji ne ugrožava njegov kvalitet.

Ove discipline imaju među sobom dosta sličnosti, ali i razlika. Ne retko se u primeni kombinuju.

5.1.1. Praksa žiga i steganografija

Poruka koja se ugrađuje u digitalni rad, može i da ne bude u vezi sa tim radom. Jasno, to onda više nije (digitalni vodeni) žig.

U praksi sakrivanja poruka u digitalne radove iskristalisale su se dve nedisjunktne discipline: *praksa žiga* i *steganografija*. Termin *steganografija* vodi poreklo iz grčkog jezika (steganos=pokriveno, graphia=pisanje) i obično se interpretira kao sakrivanje informacije u drugu informaciju. To je umetnost sakrivene komunikacije. I samo postojanje poruke je tajna.

Praksa žiga je praksa sakrivanja informacije o radu u sam taj rad. Steganografija je praksa sakrivanja tajne informacije u bezazleni sadržaj. Jasno, moguće je da se u digitalni rad sakrije i tajna poruka u vezi sa tim radom (steganografska praksa žiga), ali i da se u rad ugradi poruka koja nije u vezi sa radom, a nije ni tajna.

Često spominjan primer steganografije nalazi se u jednoj Herodotovoj priči o robu, kome je njegov gospodar Histiej (Histija) tetovirao tajnu poruku na obrijanoj glavi.

Posle tetoviranja, sačekao je da robu kosa ponovo poraste i tako sakrije poruku. Zatim ga je poslao u jonski grad Milet. Kada je rob stigao u Milet, glava mu je ponovo obrijana, tako da je upravnik grada, Aristagora, mogao poruku da pročita. Ova poruka je ohrabrila Aristagoru da digne bunu protiv persijskog kralja (Jonski ustanak).

Razlika između prakse žiga i steganografije je daleko veća nego što na prvi pogled izgleda.

Pre svega, postojanje žiga ne mora biti tajna. Vernost rada potrebna je samo iz kozmetičkih razloga. Žig treba da je neprimetan u smislu nenarušavanja estetike. Zato izvestan poremećaj u statistici može biti prihvatljiv. S druge strane, zahtev steganografije je daleko oštriji: i samo postojanje poruke predstavlja tajnu. Neprimetnost je uslov svih uslova. Neprijatelj ne sme ni da nasluti da poruka postoji.

Rad je važniji od žiga koji je u njega ugrađen. Žig postoji zbog rada, i bez njega on ne znači ništa. Ukoliko neprijatelj u pokušaju da ukloni žig upropasti rad, tako da on izgubi svoju komercijalnu vrednost, više nije bitno da li je žig očuvan ili nije. Zato se od žiga očekuje da preživi samo one intervencije koje ne ugrožavaju kvalitet rada.

S druge strane, **steganografska poruka je važnija od rada u koji je ugrađena**. Rad služi samo za sakrivanje poruke. Ali, **tajnost poruke je važnija i od same poruke**. Bilo kakve intervencije na radu treba da učine da poruka prestane da postoji. Steganografska poruka zato ne treba da je robusna. Ona treba da je *lomljiva*.

5.1.2. Steganografija i kriptografija

Kriptografija i steganografija koriste se u tajnoj komunikaciji. Često se kombinuju i zajedno koriste.

Kriptografija je disciplina koja se bavi metodama sakrivanja sadržaja informacije od neovlašćenih lica. Poruka se (en)kriptuje, tako da postaje nečitljiva za sve osim za onu osobu koja ima matematički ključ, neophodan da se poruka dekriptuje.

Kriptovanje sakriva značenje poruke koja se šalje. Osnovna ideja je da se koristi šifra koja osigurava da napadač ne može da dešifruje poruku u za njega razumnom roku. Napadač je svestan da poruka postoji, ali bez odgovarajućeg ključa ne može da odgonetne šta u njoj piše.

S druge strane, steganografija se koristi da potpuno **sakrije činjenicu postojanja tajne komunikacije**. Poruka se sakriva u naizgled bezazlen sadržaj. I samo postojanje poruke predstavlja tajnu.

Ove dve discipline često se koriste zajedno, jedno kao dopuna drugog. Poruka se kriptuje, a zatim tako kriptovana sakriva u digitalnom radu. Da bi se mogla dešifrovati steganografska kriptovana poruka, obično se moraju znati dva različita ključa – *ključ kriptovanja* i *ključ žiga* (ustvari ključ steganografske poruke).

5.1.3. Praksa žiga i kriptografija

Kriptovan digitalni rad će posle dekriptovanja biti isti kao što je bio pre kriptovanja (dekriptovanjem dobijamo rad identičan originalu). S druge strane, digitalni vodeni žig trajno menja sadržaj rada.

Kriptovanje sprečava da se rad neovlašćeno koristi. Rad može koristiti samo neko ko je platio za to. Uz pomoć ključa, rad će biti dekriptovan i moći će da se koristi. Posle dekriptovanja, rad je potpuno nezaštićen od neovlašćenog korišćenja.

Rad sa žigom može da koristi (gleda ili sluša) svako. Međutim, informacija o vlasništvu je u rad upisana trajno, sa ciljem da spreči neprijatelja da rad prisvoji (ili na neki drugi način zloupotrebi).

5.2. Primeri steganografskih tehnika

Postoji vrlo veliki skup mogućih načina sakrivanja informacije u digitalne radove. U ovom potpoglavlju navedeno je nekoliko primera realizacije steganografskih tehnika [5_01, 5_02, 5_03, 5_04].

5.2.1. Ugradnja u bitove najmanje težine (LSB pristup)

Slika u nijansama sive se u računaru predstavlja s osam bitova informacije po pikselu. Crnoj boji odgovara vrednost piksela $0 = (00000000)_2$, a beloj $255 = (11111111)_2$.

Nisu svih osam bitova u opisu piksela jednako važni. Bit najveće težine sadrži najvažniju informaciju, i njegova promena je vrlo lako uočljiva; na primer, promena ovog bita iz 0 u 1 učiniće da se boja crnog piksela promeni u srednje sivu, a boja sivog piksela u belu. S druge strane, razlika u nijansi nastala zbog promene bita najmanje težine neprimetna je za ljudsko oko.

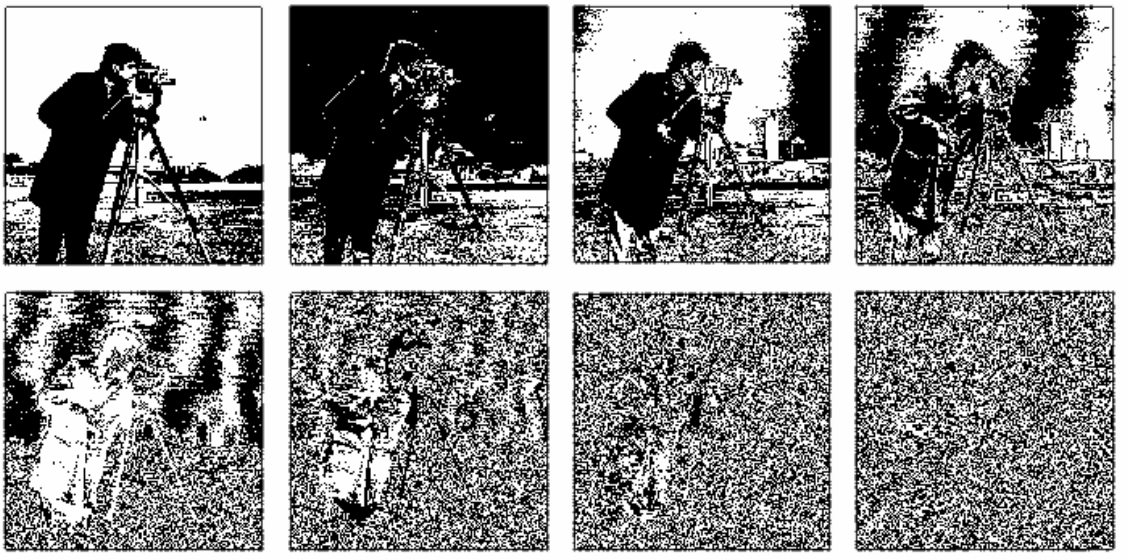
Matrica slike (dubine boje 8) može se razložiti u osam matrica slike (dubine boje 1), tako da j–ta matrica za svoje elemente ima j–te bitove slike. Ovih osam matrica nazivaju se *bit ravnima slike*. [5_04, 5_05].

Slede program za dobijanje i prikaz (slika 5.1) bitravni slike 'Cameraman'.

```
c0=double(imread('f:\zig\Cameraman.tif'));
```

```
bitravni(c0);

function K=bitravni(J);
figure
for k=1:8
    K(:,:,k)=bitget(J,k);
    subplot(2,4,9-k);imshow(K(:,:,k));
end
```



Slika 5.1: Osam bit ravni slike 'Cameraman' (poređane od bit ravni najveće, do bit ravni najmanje težine)

Bit ravan najveće težine daje grubu predstavu slike – gde je originalna slika svetla, tu je slika bit ravni bela; gde je originalna slika tamna, slika bit ravni je crna.

Bit ravan najmanje težine izgleda kao da sadrži samo šum. U steganografiji je česta praksa da se tajna informacija na neki način smesti u bitove najmanje težine. Ova praksa sakrivanja informacije obično se naziva *LSB (Least Significant Bit) pristup*.

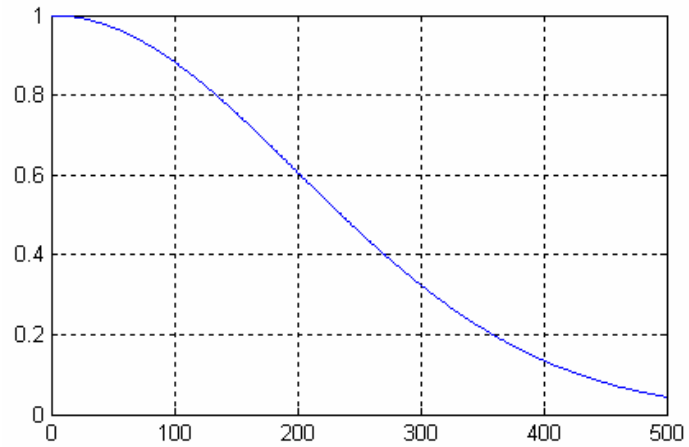
Da se ovde uneta poruka ne bi mogla lako dešifrovati (kao što bi mogla kada bi se informacija upisivala bit po bit, redom s leva na desno i odozgo na dole po matrici ove bit–ravni), dobro je krenuti od *ključa ugradnje*, popunjavanjem ove matrice bitovima, po redosledu koji je dirigovan nizom pseudoslučajnih brojeva koji se generiše polazeći od ovog ključa.

Ovde treba voditi računa i o *rođendanskom paradoksu*, tj. o činjenici da je dosta verovatno da se i kod relativno kratkog niza (pseudo)slučajnih brojeva, jedan broj pojavi više puta. Takav slučaj, poznat kao *kolizija*, učinio bi da novi unos izbriše stari (stari unos se time gubi).

Na prvi pogled, mogućnost pojave kolizije je vrlo mala. To nažalost nije tačno. Verovatnoća je mala samo za sasvim kratke poruke.

Grafik na slici 5.2 prikazuje koliko brzo se smanjuje verovatnoća da se ni jedna kolizija ne pojavi, za sliku od 200x200 piksela, u zavisnosti od broja bitova poruke. Vidi se da, već za poruku od 200 bitova (informacija se upisuje u svaki dvestoti piksel), verovatnoća da kolizije neće biti je 0.6, a za poruku od 400 bitova (svaki stoti piksel), ova verovatnoća pada na samo 0.15. To je razlog zašto se o kolizijama mora voditi računa (osim možda u slučaju sasvim kratkih poruka).

```
clear,close all %Rodjendanski paradoks
m1=500;n1=200*200;p(1)=1;
for i1=2:m1
    p(i1)=p(i1-1)*(n1-i1+1)/n1;
end
figure('Color','w'),x=1:m1;plot(x,p)
```



Slika 5.2: Grafik – ilustracija "rođendanskog paradoksa"

Da bi se prevazišao problem kolizija, dobro je pratiti indekse piksela koji su već korišćeni u komunikaciji. Ako se izabere broj koji je već korišćen, odmah se bira sledeći.

Još jedan sigurnosni dodatak ovakvoj ugradnji je da se poruka pre upisa na prethodno opisan način, kriptuje. Ovo bi iziskivalo još jedan ključ, *ključ kriptovanja* (cipher key)[22].

Ova tehnika se jednostavno realizuje. Ugrađena poruka je vrlo krhka, i nestaje i pri najmanjim uobičajenim modifikacijama slike.

5.2.2. Slika u slici

Ovo je u suštini samo još jedna varijanta LSB pristupa.

Pošiljalac tajne poruke tri bita najmanje težine originalne slike zamenjuje sa tri bita najveće težine tajne slike. Primalac izvlači tri bita najmanje težine iz primljene slike,

[22] Kriptovanje se može koristiti kao dodatak i u drugim tehnikama sakrivanja podataka.

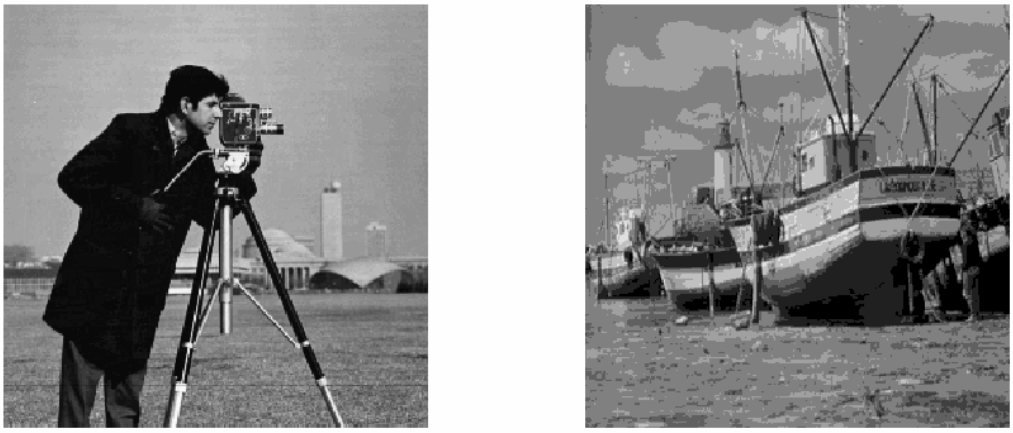
dobijajući tako pristup do tri najznačajnija bita tajne slike. Dok degradacija slike u koju je ugrađena druga slika uglavnom nije vizuelno uočljiva, tri bita je dovoljno da se pošalje gruba aproksimacija tajne slike.

Sledi program kojim su generisane slike 5.3, 5.4 i 5.5, koje ilustruju ovaj primer.

```
% Slika u slici
clear,close all
I1=imread('f:\zig\cameraman.tif');
I2=imread('f:\zig\fishingboat_m.tif');
figure ('Color','w'),subplot(1,2,1),imshow(I1),subplot(1,2,2),imshow(I2)
J1=double(I1);[d1 d2]=size(I1);J2=double(I2);
K1=bitravni(J1);K2=bitravni(J2);
Ldve=zeros(d1,d2);
ind=0
for k=8:-1:4
    Ldve=Ldve+K1(:,:,k)*(2^(k-1));ind=ind+1;
end
for k=3:-1:1
    Ldve=Ldve+K2(:,:,9-k)*(2^(k-1));ind=ind+1;
end
K=bitravni(Ldve);
Tajna=K(:,:,1)*128+K(:,:,2)*64+K(:,:,3)*32;
figure ('Color','w'),subplot(1,2,1),imshow(uint8(Ldve))
subplot(1,2,2),imshow(uint8(Tajna))
```

Slika 5.3: Originalna i tajna slika

Slika 5.4: Bit ravni slike – kombinacije nastale zamenom tri bitravni najmanje težine prve slike trima bitravnima najveće težine druge slike.



Slika 5.5: Slika – kombinacija, i slika dobijena od njene tri bit ravni najmanje težine

5.2.3. Nekorišćen ili rezervisan prostor u računarskim sistemima

Sakrivanje informacije u nekorišćen ili rezervisan prostor ne degradira digitalni rad u čiji nekorišćen prostor se ova informacija sakriva. Na primer, način na koji operativni sistemi smeštaju fajlove obično rezultuje u nekorišćenom prostoru koji je alociran za fajl. Tako, pod Windows-om XP (ili 2000), disk formatiran kao NTFS, bez kompresije obično koristi veličinu klastera od 8 KB. To znači da je minimum prostora koji se alocira za fajl 8 KB. Ako je fajl veliki 1KB, tada dodatnih 7 KB ostaje neupotrebljeno. Ovaj "dodatni" prostor može se koristiti za sakrivanje informacije bez pojavljivanja traga o tome u direktorijumu. Nekorišćen prostor u zaglavlju fajla slike ili audija može se takođe koristiti da čuva "ekstra" informaciju.

Drugi metod sakrivanja informacije u fajl sistemima je da se stvori skrivena particija. Ova particija se ne vidi ako se sistem startuje normalno. Međutim, u mnogim slučajevima, izvršavanje disk configuration utility-ja (kao što je DOS-ov ili Linux-ov FDISK) prikazuje skrivenu particiju. Ovo je dovelo do ideje da se stvori tzv. *steganografski fajl sistem*. Ako korisnik zna ime fajla i lozinku, dopušta se pristup fajlu. U suprotnom, u sistemu nema nikakvog dokaza o postojanju fajla.

5.2.4. Sakrivanje tajne poruke u tekstuelne podatke

Jedna mogućnost sakrivanja informacije u tekstuelne podatke je dodavanje "nevidljivih" karaktera tekstu. Ti karakteri bi prenosili skrivenu informaciju. HTML fajlovi se mogu koristiti da bi se u tekst uključili ekstra prostori (spaces), tabovi i oznake završetka reda. Web pregledači ne prikazuju takve prostore i redove, i njih će moći da nađe samo onaj ko gleda izvorni HTML fajl.

Načina na koje se poruke mogu sakrivati u digitalne dokumente je mnogo. Sledi opis interesantnog načina sakrivanja poruka u slike tekstuelnih podataka (skenove teksta). Ovde će biti navedena tri srodna metoda: kodiranje razmacima među redovima teksta, kodiranje prostorom između reči i kodiranje izmenama u slovima.

Kod *kodiranja razmacima među redovima*, linije teksta se neprimetno šiftuju naviše ili naniže (može se ugraditi bit 1 pomeranjem linije teksta neprimetno nagore, a 0 – pomeranjem linije teksta nadole). Za stranicu teksta sa 40 redova, na primer, to daje 2^{40} mogućih kodnih reči.

U *kodiranju horizontalnim šiftovanjem reči*, menja se prostor među rečima u redu poravnatog (justified) teksta. U skladu sa bitom tajne poruke koja se ugrađuje, menjaju se rastojanja između odabranih reči teksta u koji ugrađujemo poruku. Teoretski, moguće je menjati svako rastojanje između dve reči; jedino ograničenje je da suma svih pomeranja u svakoj liniji mora biti jednaka 0, tako da linija ostaje pravilno poravnata.

U *kodiranju karaktera*, za upis poruke mogu se koristiti male promene u slovima (na primer, može se neprimetno produžiti vrh slova "t").

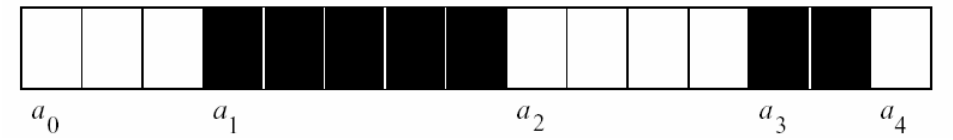
Moguće je i ova tri metoda kombinovati zajedno. Ovaj način sakrivanja informacije ima jednu manu – **poruka će nestati ako se tekst prekuca**.

Za ovakav zapis poruka vezana je jedna anegdota. 1981. godine, reprinti poverljivih dokumenata sa sastanaka britanske vlade su se sutradan po održanim sednicama pojavljivali u novinama. Priča se da je Margaret Tačer, da bi saznala ko iznosi ove informacije, na jednoj od sledećih sednica svojim ministrima podelila primerke

dokumenata koji se mogu jedinstveno identifikovati. Svaki dokument je imao različit prostor među rečima, koji je korišćen da bi se kodirao identitet primaoca. Ovo je primer *steganografske prakse žiga*, tj. tajne poruke koja je u vezi sa radom u koji je ugrađena.

5.2.5. Sakrivanje informacije u binarne slike

Binarne faks slike sadrže redundancu u načinu na koji su crni i beli pikseli raspoređeni. Prema preporuci ITU (International Telecommunication Union), faks slike se mogu kodirati koristeći kombinaciju RL (run length) i Hafmenovog kodiranja. RL tehnike koriste činjenicu da u binarnoj slici uzastopni pikseli vrlo verovatno imaju istu boju. Slika 5.6 pokazuje jednu sken liniju faks dokumenta. Pozicije sa promenom boje označene su sa a_i.



Slika 5.6: Jedna sken linija binarne slike

Umesto da se boja svakog piksela kodira eksplicitno, RL metodi kodiraju pozicije promene boje (a_i) zajedno sa brojem $RL(a_i, a_{i+1})$ uzastopnih piksela koji imaju istu boju počevši od a_i. Hipotetična sken linija sa slike bi se kodirala sa $\langle a_0,3\rangle,\langle a_1,5\rangle,\langle a_2,4\rangle,\langle a_3,2\rangle,\langle a_4,1\rangle$. Tako se binarna linija može opisati sekvencom RL elemenata $\langle a_i, RL(a_i, a_{i+1})\rangle$.

Informacija se može ugraditi u binarnu, RL kodiranu sliku modifikovanjem poslednjeg značajnog bita od $RL(a_i, a_{i+1})$. U procesu kodiranja modifikuju se dužine sekvenci binarne slike tako da je $RL(a_i, a_{i+1})$ parno ako je i–ti bit tajne poruke, $m_i = 0$ (a neparno ako je $m_i = 1$). To se može postići, na primer, na sledeći način: Ako $m_i = 0$, a $RL(a_i, a_{i+1})$ je neparno, pozicija od a_{i+1} se seli jedan piksel levo (ako je $m_i = 1$, a $RL(a_i, a_{i+1})$ parno, a_{i+1} se pomera jedan piksel udesno).

5.2.6. Funkcije imitiranja

U ogromnoj količini informacija koje se šalju preko mreže, ljudi ne mogu sami da prate svu komunikaciju. Zato službe koje pokušavaju da kontrolišu protok informacija sve više koriste automatske sisteme za nadzor. Ovi sistemi kontrolišu komunikaciju praćenjem ključnih reči i statističkog profila poruke. Na primer, zahvaljujući različitim

statističkim svojstvima, moguće je automatski razlikovati kriptovane od nekriptovanih poruka.

Wayner [5_06] je predložio funkcije imitiranja[23] koje menjaju statistički profil poruke, tako da odgovara statističkom profilu običnog nekriptovanog teksta.

Ipak, ove funkcije mogu da prevare samo mašine. Čovek će vrlo brzo videti da su ovako stvoreni tekstovi potpuno besmisleni i da su puni gramatičkih i tipografskih grešaka.

Nova generacija funkcija imitiranja zato omogućuje da se uz pomoć kontekstno slobodnih gramatika, stvaraju tekstovi koji podsećaju na tekstove uobičajene komunikacije.

[23] *Funkcija imitiranja* menja fajl A tako da on poprimi statistička svojstva drugog fajla B. Tako, ako je $p(t, A)$ verovatnoća da se neka niska t pojavi u A, tada funkcija imitiranja f prekodira A tako da je $p(t, f(A))$ približno sa $p(t, B)$ za sve niske t dužine manje od nekog n.

6. Svojstva žiga

Ovo poglavlje zasnovano je pre svega na sadržaju knjige [4_01]. O svojstvima žiga može se naći i u drugim tekstovima, na primer u [6_01, 6_02].

6.1. Punjenje

Punjenje žiga se za zvučni fajl definiše kao broj bitova informacije žiga u jedinici vremena. Za video, to je broj bitova informacije po jedinici vremena, ili po frejmu. Punjenje žiga za slikovni fajl definiše se kao broj bitova informacije po fajlu.

To je parametar koji direktno utiče na vernost rada i robusnost žiga. Što se više informacije ugrađuje u rad, to su manje vernost i robusnost.

Za razliku od primetnih žigova ili poruka koje ne moraju da budu robusne, kod neprimetnog a robusnog žiga, količina informacije koja se ugrađuje vrlo je mala, a zavisi od aplikacije. To je u svakom slučaju najmanja količina potrebna da se izvrši identifikacija za koju je žig namenjen.

U aplikaciji kontrole kopiranja za video, dovoljno je da postoji bit informacije na svakih nekoliko sekundi. Kod kontrole emitovanja, informacija je duža – treba jedinstveno da odredi rad koji se emituje. U aplikaciji praćenja transakcija, informacija je još duža da bi jedinstveno identifikovala sve učesnike u transakciji.

Informacija koja se ugrađuje u digitalni rad radi zaštite autorskih prava, treba jedinstveno da identifikuje nosioca prava. To može biti nešto kao:

ISBN (International Standard Book Number), desetocifreni broj koji jedinstveno određuje svaku štampanu knjigu;

ISRC (International Standard Recording Code), koji jedinstveno označava audio ili video fajl. Sastoji se od: koda države (2 ASCII karaktera), koda vlasnika (3 ASCII karaktera), godine snimanja (2 cifre), serijskog broja (5 cifara).

Uz to će verovatno biti dodata godina zaštite autorskih prava, dozvole i možda još neki podaci. To znači da ugrubo 70 bitova informacije treba da bude ugrađeno u rad, ako je

aplikacija žiga zaštita autorskih prava. Ako se poruci dodaju podaci kao što su digitalna signatura, kôd provere parnosti, kodovi korekcije greške,..., poruka može da postane nešto duža, ali obično ne preko nekoliko stotina bitova [6_01].

6.2. Greške u detekciji

Detektor izveštava da li je u digitalni rad ugrađen žig (i kakvu informaciju nosi). On tu može da napravi jednu od tri moguće greške:

- Da izvesti da rad sadrži žig, iako u njega nikakav žig nije ugrađen (*greška lažnog pozitivnog*)
- Da izvesti da rad ne sadrži žig, iako je žig ugrađen (*greška lažnog negativnog*)
- Da tačno izvesti da žig postoji, ali da pogrešno izvesti koja je poruka ugrađena (*greška poruke*)

U radu sa digitalnim žigovima, sve navedene greške mogu da se pojave. Cilj je da se pojavljuju dovoljno retko, a da vernost digitalnog rada ostane zadovoljavajuća.

6.3. Robusnost žiga

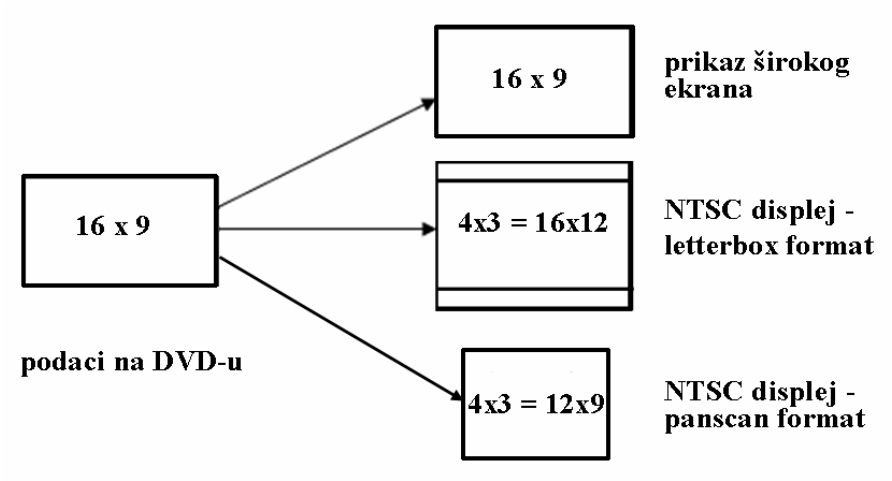
Žig je ugrađen *efikasno* ako ga je moguće detektovati neposredno po ugradnji.
Jedan od osnovnih zahteva koje žig treba da zadovolji je *robusnost* – žig treba da je u stanju da preživi uobičajene operacije za koje je verovatno da se pojave od momenta ugradnje, do detekcije.
Teško je predvideti koje sve operacije će rad pretrpeti. Praviti žig koji će preživeti sve moguće operacije teško je i skupo, ako ne i nemoguće. Treba naći meru, i **napraviti žig koji će preživeti operacije za koje je najverovatnije da će se pojaviti**, a koje pri tome zadržavaju kvalitet rada.

6.3.1. Sprečiti...

Neke od strategija za stvaranje robusnih žigova pokušavaju da naprave žig koji će uspeti da preživi očekivane intervencije nad radom.
U ovu klasu rešenja spada *redundantna ugradnja*. Žig se ugrađuje u rad višestruko (ponekad na različite načine), sa nadom da će neka od ugradnji preživeti izobličenje. Primer za ovo je ugradnja žiga belog šuma, koji bit informacije ugrađuje redundantno u prostornom domenu. Ako neko sliku opseče, u preostalom delu slike ostaće dovoljno informacije da se žig može detektovati.

Ako se očekuje da će slika biti izložena JPEG kompresiji, može se primeniti neka metoda ugradnje žiga u blokove veličine 8×8 u domenu diskretne kosinusne transformacije, na način koji će omogućiti da žig preživi ovakvu kompresiju. U literaturi je predloženo više metoda ugradnje u DCT koeficijente, i o ovome će biti više reči nešto kasnije.

Ponekad, postoji nekoliko dobro definisanih očekivanih izobličenja. Takve situacije se često rešavaju prethodnim invertovanjem budućih izobličenja u ugrađivaču. Primer koji sledi naveden je u [4_01] u vezi sa praksom žiga u video domenu, preciznije, u vezi sa načinom na koji DVD rukuje sa odnosom širine i visine slike (aspect ratio). Standardne NTSC i PAL televizije imaju aspect ratio od 4:3; high–definition televizija (HDTV) ima ovaj odnos 16:9. DVD diskovi čuvaju celu 16:9 sliku, a nude dva metoda za stiskanje slike na standard 4:3 (=12:9=16:12) za TV ekrane. U *letterbox* formatu slika se smanjuje da se uklopi u širinu formata ekrana (4:3), sa nešto crnih linija dodatih gore i dole. U *panscan* formatu slika se uklapa u visinu ekrana i saseca s leva i s desna (slika 6.1).



Slika 6.1: Tri displej moda za DVD

Žig za DVD mora biti ugrađen u izvor širokog ekrana (16:9), i moguće je detektovati ga u prikazu na širokom ekranu. Međutim, letterbox i panscan modovi uvode izobličenja koja će verovatno učiniti da se ovaj žig ne detektuje. Tako, ovde postoji mali broj (dva) izobličenja kojima će slika i žig verovatno biti izloženi.

Rešenje je da se ugrade tri različita žiga u rad, tj. žig za svaki od navedena tri slučaja. Prvi žig će biti detektovan ako nije bilo nikakve konverzije formata. Drugi i treći žig se pre ugradnje (na ugrađivaču) izlažu inverziji budućeg izobličenja, tako da pri emitovanju (letterbox ili panscan format) mogu biti detektovani.

6.3.2. ... i lečiti

Često nije moguće da se tačno predvidi koje operacije će slika i žig pretrpeti. Ponekad zato u detektor žig stigne izobličen, takav da ga nije moguće odmah detektovati.

U tom slučaju, u potrazi za žigom treba naslutiti koje intervencije su se od ugradnje do detekcije dogodile. Kada se odredi kojim izobličenjima je slika bila izložena, pokušava se da se ova izobličenja invertuju, da bi se žig mogao detektovati. Primer je rotacija slike. Ako se otkrije ugao (φ) za koji je slika rotirana, moguće je izobličenje invertovati, rotiranjem za $-\varphi$.

Težak korak u invertovanju izobličenja u detektoru je da se odredi kom izobličenju je slika bila izložena. To ponekad iziskuje da se obavi iscrpno pretraživanje, koje će pomoći da se dobije odgovor na to pitanje. Na primer, detektor žiga bi rotirao sliku za svaki od uglova od 0^o do 359^o sa korakom od 1^o. U nekim slučajevima nije potrebno ispitati ovako veliki broj mogućnosti, nego je moguće unapred identifikovati mali broj kandidata izobličenja za koje je verovatno da su se mogla desiti.

6.4. Robusnost na različite klase izobličenja

6.4.1. Aditivni šum

Neki procesi kojima se rad izlaže imaju efekat dodavanja slučajnog signala:

$c_n = c + n$

gde je c rad, a n slučajni vektor izabran iz neke raspodele, nezavisno od c. Na primer, audio emisija preko radio kanala može biti oštećena belim šumom, što će rezultovati u smetnjama. Slično, video emisija preko TV kanala može pokupiti video sneg. U tim slučajevima, šum je nezavisan od rada. Takvi šumni procesi su slučajevi aditivnog šuma.

Veliki broj tehnika ugradnje žiga metodom raširenog spektra, zasniva se na dodavanju aditivnog šuma radu.

6.4.2. Promene amplitude

Većina procesa primenjenih na radove sa žigom ne modelira se dobro aditivnim šumom. Pretrpljene promene obično su korelirane sa radom. Čak, mnogi procesi su determinističke funkcije rada. Važan primer je promena u amplitudi:

$c_n = \nu c$

c je originalan rad, a v faktor skaliranja. Za zvuk, to je promena u jačini zvuka. Za sliku, to je promena u sjajnosti i kontrastu.

6.4.3. Kompresija (sa gubicima)

U kompresiji sa gubicima, deo informacije se odbacuje radi uštede u memorijskom prostoru. Mada je ovaj gubitak informacije trajan, on je obično prihvatljiv, jer računarski prikaz signala sadrži redundancu u odnosu na ono što je potrebno ljudskom opažanju.

Između prakse žiga i kompresije sa gubicima postoji suštinski konflikt. Idealna kompresija sa gubicima treba da odbaci svu suvišnu informaciju, i sve perceptualno ekvivalentne radove prevede u jedinstven komprimovan prikaz. To znači da se posle idealne kompresije sa gubicima ne može desiti da dva perceptualno identična rada imaju dva različita komprimovana prikaza. Drugim rečima, **kompresija pokušava da ukloni svu psihovizuelnu redundancu iz sadržaja, a žigovi pokušavaju da kodiraju informaciju u toj redundanci**.

S druge strane, žig treba da preživi kompresiju. Zato rad sa žigom i isti rad bez žiga posle kompresije ne bi izgledali isto, tj. vernost i robusnost žiga kod idealne kompresije sa gubicima se isključuju.

Na sreću, u praksi, algoritmi kompresije su daleko od idealnih i još uvek je razumno jednostavno da žig u radu preživi kompresiju sa gubicima, uz i dalje odličnu vernost rada.

6.4.4. Robusnost na vremenska i geometrijska izobličenja

Ovo je jedna od najtežih, i zasad još nerešenih oblasti u istraživanjima prakse žiga. Robusnost u odnosu na njih je trenutno velika oblast istraživanja.

Sva geometrijska izobličenja slike mogu da se izraze sa

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{6.1}$$

gde $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ predstavlja neizobličenu, a $\begin{bmatrix} x_n \\ y_n \end{bmatrix}$ – izobličenu lokaciju piksela.

U slikovnim (i video) podacima, ovoj klasi pripadaju različita izobličenja, među kojima su i translacija, rotacija, i promena dimenzije slike.

Translacija je kompletno predstavljena vektorom $\begin{bmatrix} x_t \\ y_t \end{bmatrix}$. Matrica $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ se koristi da definiše sve druge afine transformacije:

Skaliranje se opisuje matricom $\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$, sa promenom odnosa širine i visine (*aspect ratio*) ako je $s_x \neq s_y$.

Rotacija za ugao θ data je matricom $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$.

6.5. Sigurnost žiga

Sigurnost žiga se definiše kao njegova sposobnost da se odupre neprijateljskim napadima.

Postoje tri osnovna nivoa napada na žigove:

- neovlašćena detekcija
- neovlašćena ugradnja
- neovlašćeno uklanjanje.

Neovlašćena detekcija predstavlja *pasivni*, a neovlašćeni ugradnja i uklanjanje *aktivne napade* na svrhu žiga.

U praksi žiga, neko mora imati mogućnost da detektuje, ugradi i/ili ukloni žigove, dok drugi moraju biti ograničeni u izvršavanju nekih, ili svih, od tih akcija.

U odlučivanju ko koju operaciju sme da obavi, svet se deli na grupu poverljivih pojedinaca (to su obično ljudi kojima žig treba da koristi) i javnosti (za koju se smatra da su potencijalni neprijatelji).

6.5.1. Neautorizovana detekcija

Najkompletniji oblik neautorizovane detekcije dešava se kada neprijatelj uspe da dekodira poruku žiga. Da bi se neprijatelj sprečio da realizuje ovaj tip napada na žig, često se sa ugradnjom žiga kombinuje kriptovanje – poruka se prvo kriptuje, pa tek onda ugrađuje.

Nekad je, međutim, neprijatelju dovoljno i samo saznanje da li postoji poruka ugrađena u rad. Mada je ovaj napad, pre svega, problem steganografije, postoje i druge situacije u kojima je znanje o poruci neprijatelju dovoljno. Na primer, neprijatelju će mogućnost detekcije predstavljati veliku prednost u pokušaju uklanjanja žiga.

Treći tip neautorizovane detekcije nalazi se negde između navedena dva. To je slučaj kada neprijatelj može da pravi razliku među oznakama koje kodiraju različite poruke, čak i ako ne zna šta poruke znače. Znači, za dva rada sa žigovima, on zna da li žigovi kodiraju iste ili različite poruke. Ovaj tip napada je problem ako neprijatelj može na neki drugi način, bez dekodiranja poruka, da sazna njihovo značenje.

6.5.2. Neautorizovana ugradnja

Najkompletniji tip neautorizovane ugradnje dešava se kada neprijatelj sastavi i ugradi svoju originalnu poruku.

Manje kompletan tip neautorizovane ugradnje pojavljuje se kada neprijatelj na neki način nabavi žig iz nekog rada, i ugradi ga nezakonito (umesto nove poruke) u drugi rad. Pri tome on čak ne mora da zna kako je poruka kodirana. Interpretacija za ovakav *napad kopiranja* je da je ovako **ugrađena poruka** u drugi rad **validna, ali je rad** u koji je ugrađena **pogrešan**. Kako žigovi po definiciji nose informaciju o radovima u koje su ugrađeni, poruke koje oni nose ne mogu biti pravilno interpretirane bez upućivanja na rad. Poruka "Ovo pripada..." (ili: "Ovo smeš da kopiraš") odnosi se na određen, a ne na svaki digitalni rad. Zato rad u koji se poruka ugrađuje mora biti posmatran kao implicitni deo poruke.

Jedan drugi tip neautorizovane ugradnje je čuveni *Krejverov (Craver) napad* [6_03, 6_04, 6_05] (u literaturi poznat i pod mnogim drugim imenima: *IBM-ov napad, napad dvosmislenošću, falsifikovanjem, ćorsokakom, inverzijom*). On je izazvao pravu pometnju među ljudima koji se bave praksom žiga. Nije ni čudo, jer dovodi u pitanje svrhu korišćenja žigova. Naime, u najatraktivnijoj primeni, dokazu vlasništva u slučaju sudskog spora, tehnika podložna ovom napadu je neupotrebljiva.

Uobičajeni algoritam ovog napada može se predstaviti sledećim scenarijem.

Alisa[24] ugrađuje poruku S_A u originalnu sliku $\boldsymbol{c}_0$, pomoću funkcije ugradnje ε_A. Rezultujuća slika sa žigom je $\boldsymbol{c}_w$:

$$\varepsilon_A(c_0, S_A) = c_w \tag{6.2}$$

Bob koristi funkciju ugradnje ε_B', i u sliku $\boldsymbol{c}_w$ ugrađuje neku poruku žiga S_B'. Rezultujuća slika je $\boldsymbol{c}'$:

[24] Alisa i Bob su već uobičajeni likovi u opisivanju napada u kriprografiji i praksi digitalnog vodenog žiga.

$$\varepsilon_B'(\boldsymbol{c}_w, \boldsymbol{S}_B') = \boldsymbol{c}' \tag{6.3}$$

Ako je Bobova funkcija ugradnje ε_B' takva da za nju postoji "inverz" ε_B, čijom se primenom na sliku $\boldsymbol{c}'$, ugradnjom neke poruke S_B dobija slika $\boldsymbol{c}_w$:

$$\varepsilon_B(\boldsymbol{c}', \boldsymbol{S}_B) = \boldsymbol{c}_w, \tag{6.4}$$

onda Bob može, sa istim argumentima kao Alisa, da tvrdi da je original njegova slika $\boldsymbol{c}'$, a da je slika $\boldsymbol{c}_w$ nastala iz nje ugradnjom poruke S_B, funkcijom ugradnje ε_B.

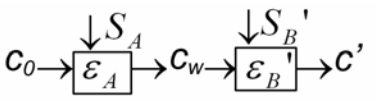
U ovim okolnostima, kao što Alisa može uz pomoć svog detektora određenog funkcijom δ_A da dokaže prisustvo poruke S_A u slikama $\boldsymbol{c}_w$ i $\boldsymbol{c}'$:

$$\delta_A(\boldsymbol{c}_w) = \boldsymbol{S}_A, \qquad \delta_A(\boldsymbol{c}') = \boldsymbol{S}_A \tag{6.5}$$

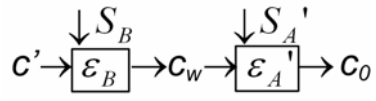
isto tako Bob može da dokaže prisustvo poruke žiga S_B u slikama $\boldsymbol{c}_w$ i $\boldsymbol{c}_0$:

$$\delta_B(\boldsymbol{c}_w) = \boldsymbol{S}_B, \qquad \delta_B(\boldsymbol{c}_0) = \boldsymbol{S}_B \tag{6.6}$$

Tako sud neće moći da odgovori na pitanje da li je redosled nastanka slika



ili



Mana algoritma ugradnje koji je podložan napadu falsifikovanja je *invertibilnost* (u ovom primeru postoje slike $\boldsymbol{c}_w$ i $\boldsymbol{c}'$ koje tim algoritmom mogu nastati jedna iz druge – ne zna se koja je original, a koja kopija sa žigom).

Krejver i njegovi koautori su u navedenim radovima ustvrdili da je najveći broj poznatih algoritama žiga, ako ne i svi, podložan ovom napadu.

6.5.3. Neautorizovano uklanjanje

Žig se smatra uklonjenim iz rada ako je postalo nemoguće detektovati ga. Rad time uopšte ne mora biti restaurirani original. Dovoljno je da liči na original, a da ne pobuđuje detektor žiga.

Razlikuju se dva nivoa napada uklanjanja. *Napadi eliminacije* uništavaju žigove tako da ih više nikakav detektor ne može naći. *Napadi maskiranja* ne uklanjaju stvarno žig; oni samo čine da je za detekciju žiga potreban detektor sofisticiraniji od postojećeg. Primer je blago rotiranje slike, koje će učiniti da detektor ne može da detektuje žig. Detektor bolji, sofisticiraniji od postojećeg, moći će da invertuje rotaciju i detektuje žig.

O napadima uklanjanja napisano je mnogo radova [6_06, 6_07]. U borbi sa takvim napadima, kao oružje se pojavljuju i *benchmarking* alatke – programi koji ispituju sposobnosti žiga da odole neprijateljskom napadima. Među takvim alatkama svakako je

najpoznatiji *Stirmark* [6_08]. On za datu sliku sa žigom generiše veći broj modifikovanih slika koje mogu biti korišćene za verifikovanje da li ja žig u stanju da odoli određenim napadima.

Između robusnosti žiga i sigurnosti prema neprijateljskim napadima uklanjanja razlika je u osnovi u tome da u drugom slučaju postoji loša namera. Mada postoje i specijalizovane tehnike uklanjanja (na primer *Jitter* i *Mosaic* napadi – [6_06]), neprijatelj može pokušati da odstrani žig i koristeći uobičajene operacije (na primer kompresiju sa gubicima[25] ili promenu kontrasta na slici). Zato će u toku daljeg teksta, pri analizi robusnosti žiga, ponekad **i takve uobičajene operacije biti nazivane napadima (uklanjanja).**

6.6. Vernost i kvalitet

Vernost je mera sličnosti između signala pre i posle obrade. Reprodukcija visoke vernosti je reprodukcija koja je vrlo slična originalu. Reprodukcija niske vernosti razlikuje se od originala.

Očuvanje vernosti rada pri ugradnji je jedan od osnovnih zahteva koji se pojavljuju u praksi žiga. Međutim, za razliku od steganografije, gde posmatrač ne sme ni da nasluti da poruka postoji, zahtevi koje rad sa žigom mora da zadovolji nisu toliko strogi. Ustvari, ovde se obično održava neki određen, zadovoljavajući nivo vernosti, a da je pri tome zadržana robusnost.

U steganografiji, osim neprimetnosti za slučajnog posmatrača, vodi se računa i o statističkoj neprimetnosti – postojanje poruke ne sme da se nasluti ni iz činjenice da su statističke karakteristike rada promenjene. Kod žiga se ne mora biti do te mere isključiv – statističke promene (kao na primer povećana entropija dela rada ili celog rada) mogu biti dopušteni.

Postoje aplikacije žiga kod kojih, u trenutku predstavljanja rada javnosti, original (slika pre obrade) nije raspoloživ. Tada se umesto vernosti procenjuje kvalitet rada.

Kvalitet je apsolutna mera dopadanja. Visokokvalitetna slika ili melodija prosto izgleda ili zvuči dobro. To znači da nema vidljivih artifakata obrade. Takav rad ne mora uopšte biti verna kopija originala. Na primer, slika kojoj su podešeni sjajnost i kontrast je dosta

[25] U daljem tekstu, 'kompresija sa gubicima' će se kraće nazivati samo 'kompresija'

različita od originala, a može od njega izgledati i dosta bolje (biti višeg kvaliteta). To je primer visokog kvaliteta, a niske vernosti.

U nekim aplikacijama važnije je da se pri ugradnji žiga očuva vernost, a u drugim je primaran kvalitet. Na primer, umetnik – autor slike može da zahteva da ugradnja žiga bude takva da slika ostane naizgled potpuno nepromenjena. Isto važi i za medicinske snimke. S druge strane, u situaciji u kojoj se ne može pristupiti originalu radi poređenja, vernost se i ne može vrednovati, i tu je bitno da se zadrži što viši kvalitet.

Vernost rada sa žigom meri se poređenjem sa radom bez žiga. Pravilo je da se u momentu merenja vernosti oba rada **razlikuju jedino u tome što jedan od njih ima žig, a drugi ga nema**.

Tako, poređenje se može obaviti neposredno po ugradnji. No, mada je ovo uvek dobro uraditi, još je važnije **uporediti ova dva rada u trenutku predstavljanja javnosti**.

Ako je rad sa žigom od ugradnje do detekcije preživeo neko izobličenje (na pr. promenu sjajnosti i kontrasta, rotaciju ili kompresiju sa gubicima), vernost se procenjuje njegovim poređenjem sa istim radom bez žiga, koji je preživeo to isto izobličenje.

Vernost (a ni kvalitet) nije binarni uslov. Izmenjeni rad je više ili manje veran originalu. Takođe, on je više ili manje kvalitetan. Razvijene su mnoge tehnike za procenu nivoa vernosti i kvaliteta.

Merenje vernosti i kvaliteta digitalnih radova nije jednostavno. Postoje dva tipa procene. Kod *subjektivne procene* koriste se ljudi – posmatrači; kod *objektivne procene* vrše se električna merenja.

6.6.1. Subjektivno vrednovanje

Kako su digitalni radovi namenjeni posmatranju od strane ljudi, potrebno je da promene u njima procenjuju ljudi. U studijama koje koriste sud ljudi, vodi se računa i o činjenici da vizuelni i slušni osećaji variraju od pojedinca do pojedinca. Ovi osećaji se čak i za jednog pojedinca menjaju u vremenu. Zato studije koje uključuju ljudsko vrednovanje, koriste veliki broj ljudi koji obavljaju veliki broj posmatranja.

U ljudskom vrednovanju vernosti i kvaliteta, kao jedinica izobličenja koristi se *JND* (*just noticeable difference*, jedva primetna razlika). JND se definiše kao nivo izobličenja koji može biti uočen u 50% posmatranja. Ova jedinica se obično smatra za minimum koji je generalno primetan.

Višestruki JND se u literaturi obično definiše sa $2\ JND = 1\ JND \quad od \quad 1\ JND$

Uobičajeni test *kvaliteta* slika ljudskim vrednovanjem obavlja se tako što se posmatračima predstavljaju dve po dve slike za koje oni treba da kažu koja je original, a koja kopija (pretpostavka je da je original kvalitetniji od kopije). U ovom eksperimentu se smatra da je pogađanje od 50% – slučajno pogađanje. Zato, ako posmatrači razliku uoče u 75% slučajeva, takva razlika se definiše kao 1 JND.

Eksperiment koji se često koristi u ispitivanju vernosti sastoji se od grupa od po tri rada, od kojih je jedan označen kao original, a od preostala dva, jedan je istovetan sa originalom, a drugi sasvim malo promenjen. Posmatrač treba da zaključi koji od ta dva rada se razlikuje od originala.

U subjektivnom merenju vizuelnog kvaliteta i vernosti, od posmatrača se traži da vrednuju slike korišćenjem skale, usvojene standardom ITU-R Rec.500 [6_09]. To je skala sa pet ocena:

- Oštećenja su neprimetna
- Oštećenja su jedva primetna
- Oštećenja su primetna, ali nisu neugodna
- Oštećenja su neugodna
- Oštećenja su krajnje neugodna

Subjektivno vrednovanje vernosti i kvaliteta je skupo i nije ga lako ponavljati. Iziskuje veliki broj radova i posmatrača. Vrednovanje dugo traje jer se oči brzo zamore. Alternativni pristup je korišćenje automatizovanog (objektivnog) vrednovanja.

6.6.2. Objektivno vrednovanje

U praksi je vrlo teško naći objektivnu meru vernosti, a još je teže naći objektivnu meru kvaliteta rada. Ipak, najčešće i nije neophodno da se predvide tačni rezultati testova koji se dobijaju ljudskim vrednovanjem. Obično je dovoljno predvideti relativno izvršenje u tim testovima.

Objektivna mera vernosti rada sa žigom predstavlja se nekom funkcijom $D(c_0, c_w)$, koja određuje rastojanje između originalnog rada c_0 i rada sa žigom, c_w. Jedna od najjednostavnijih funkcija rastojanja je *funkcija srednje kvadratne greške* (mean square error, MSE):

$$MSE(c_0, c_w) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left[c_w(i,j) - c_0(i,j) \right]^2 \tag{6.7}$$

(radovi c_0 i c_w slike dimenzija $m \times n$).

MSE se često koristi kao grubi test uticaja ugradnje žiga na vernost, iako se zna da daje slabu procenu stvarne vernosti. Mera vernosti kao što je MSE može da greši bilo podcenjivanjem, bilo precenjivanjem uočljivosti razlike između dva rada.

Tako, u slučaju slike šiftovane za jedan piksel, MSE precenjuje perceptualno rastojanje. Slika se gotovo ne može razlikovati od originala, a MSE je vrlo veliko.

U objektivnoj proceni vernosti često se koristi i *odnos signal–šum* (signal–to–noise ratio, SNR):

$$SNR(c_0, c_w) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[c_0(i,j)]^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}[c_w(i,j) - c_0(i,j)]^2} \tag{6.8}$$

Što je veći SNR, viša je vernost obrađene slike, c_w, tj. obrađena slika c_w je bliža originalnoj slici c_0. Međutim, HVS ne odgovara na vizuelni nadražaj na jednostavan način. SNR nam ne pruža uvek pouzdane ocene vernosti slike.

Objektivno vrednovanje ne daje uvek dobru ocenu vernosti slike. S druge strane, njegova primena je daleko brža i jednostavnija od subjektivnog vrednovanja. Osim toga, testovi objektivnog vrednovanja mogu se po potrebi ponavljati. Zbog ovih prednosti, objektivna procena se široko koristi.

U praksi je sve prisutnija tendencija da se subjektivna i objektivna procena kombinuju.

6.7. Važnost poznavanja ponašanja ljudskog vizuelnog sistema

Ljudski vizuelni sistem (HVS) opaža spoljni svet na vrlo komplikovan način. Njegov odgovor na vizuelne nadražaje nije linearna funkcija intenziteta nekih fizičkih nadražaja, kao što su intenzitet i nijansa. U HVS, vizuelna informacija se ne opaža jednako; neka informacija može biti važnija od druge.

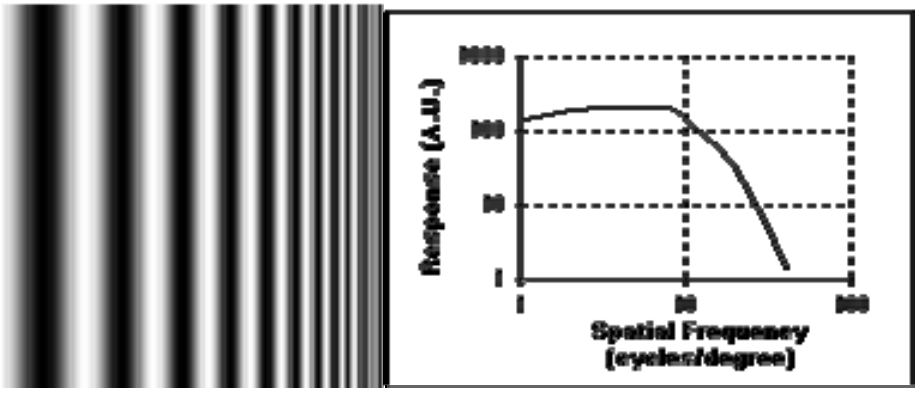
Da bi se podaci žiga što neprimetnije smeštali sliku, treba dobro poznavati svojstva HVS. Tako će se bolje moći predvideti njegove reakcije na promene u slici.

Tako, očekuje se odgovor na pitanje koje komponente u slici su perceptualno beznačajne (ispod praga vidljivosti). U praksi žiga, rezultati ovih istraživanja mogu se koristiti kao baza za algoritme ugradnje žiga, za identifikaciju komponenti koje mogu biti neprimetno zamenjene podacima žiga.

6.7.1. HVS i frekvencija

U viziji, proučavaju se odgovori HVS na tri različita vida frekvencije. To su prostorna, spektralna i vremenska frekvencija.

Prostorne frekvencije čovek zapaža kao šare ili teksture. U proučavanju osetljivosti oka na promene sjajnosti u funkciji prostorne frekvencije, došlo se do zaključka da je ljudsko oko najosetljivije na promene sjajnosti u frekvencijama srednjeg ranga (prostorna frekvenca od oko 10 ciklusa po stepenu). Osetljivost opada sa nižim i višim frekvencijama (slika 6.2).

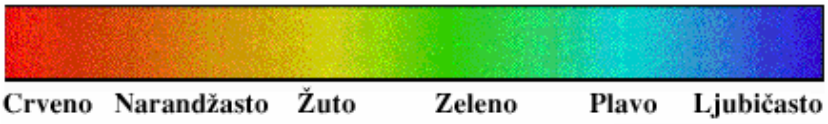


Slika 6.2: Osetljivost na prostornu frekvenciju

Tako, ako je monitor računara udaljen od očiju 50 cm (1° na 50 cm znači oko $50tg(1^\circ) \approx 0.87cm$ na ekranu), onda je prostorna frekvenca maksimalnog odgovora oka $f_{\max} = 10\, ciklusa / 0.87\ cm = 11.46\, ciklusa / cm$.

Kod slika (dvodimenzioni objekti), osetljivost oka nije zavisna samo od frekvencije različitih obrazaca nego i od njihovih orijentacija. Oko je najosetljivije na vertikalne i horizontalne linije i ivice na slici, a najmanje je osetljivo na linije i ivice sa orijentacijom od 45°.

Spektralne frekvencije se zapažaju kao *boje* (slika 6.3). Oblast osetljiva na boje u HVS sastoji se od tri zasebna skupa *čepića*; svaki skup je osetljiv na svetlost jedne od tri osnovne boje: crvene (R), zelene (G) i plave (B). Zato se svaka boja koju zapazi HVS može posmatrati kao linearna kombinacija tri osnovne boje.



Slika 6.3: Spektar vidljive svetlosti

Međutim, ljudsko oko ne reaguje u jednakoj meri na sve boje, jer u oku ima mnogo manje čepića B, nego G i R. Ono najbolje primećuje boje "srednjeg ranga", kao što su

žute i zelene, a slabije boje na kraju spektra, crvene i (najslabije) plave. Jednak prikaz crvene, zelene i plave dovodi do neefikasnog prikaza podataka kada je krajnji gledalac HVS. U nekoliko sistema ugradnje žiga u slike u boji stavlja se najveći deo signala žiga u plavi kanal RGB slike (jer je tu ljudsko oko najmanje osetljivo).

Osim toga, HVS je mnogo osetljiviji na komponentu sjajnosti nego na nijanse boje.

Vremenske frekvencije oko zapaža kao *pokret*. Osetljivost oka rapidno opada za frekvencije iznad 30 Hz. To je razlog zašto se TV i filmske slike ne smenjuju brže od 60 frejmova u sekundi.

6.7.2. Maskiranje svetlošću, teksturom i frekvencom

Fenomen *maskiranja* predstavlja dejstvo na ljudska čula jednog, **u prisustvu drugog nadražaja**.

Ovo je fenomen koji je odavno izučavan. O njemu govori i Weber–Fechner-ov zakon – pokušava da opiše vezu između fizičkog i opaženog intenziteta nadražaja.

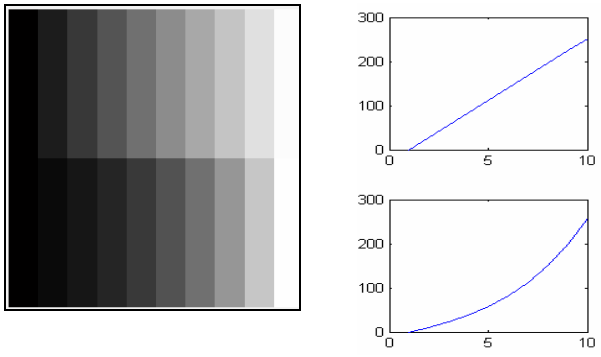
Intenzitet subjektivne senzacije proporcionalan je logaritmu intenziteta nadražaja.

Ovo važi za praktično sva ljudska čula, pa i za čulo vida. Oko oseća svetlost logaritamski.

Posledica: Oko bolje primećuje promene u tamnijim nego u svetlijim delovima slike.

Jednako opaženi koraci u sjajnosti, zahtevaju da fizička sjajnost (nadražaj) raste eksponencijalno. To se vidi na sledećem primeru (slika 6.4).

```
Matrica piksela:
0 28 56 84 112 140 168 196 224 252
0 28 56 84 112 140 168 196 224 252
0 28 56 84 112 140 168 196 224 252
0 28 56 84 112 140 168 196 224 252
0 28 56 84 112 140 168 196 224 252
0 10 22 38  57  82 112 150 198 255
0 10 22 38  57  82 112 150 198 255
0 10 22 38  57  82 112 150 198 255
0 10 22 38  57  82 112 150 198 255
0 10 22 38  57  82 112 150 198 255
```



Slika 6.4: Ilustracija Weber–Fechner-ovog zakona

Gornji deo matrice i slike pokazuje linearni porast objektivne, ali logaritamski porast subjektivne sjajnosti (čini nam se da sjajnost sve sporije raste). Donji deo matrice i slike pokazuje eksponencijalni porast objektivne, a linearni porast subjektivne sjajnosti (čini

nam se da sjajnost raste linearno). Desno su grafici porasta objektivne sjajnosti za piksele gornjeg, odnosno donjeg dela matrice i slike.

U svetlijim oblastima manje se vide promene nego u tamnijim. Ako je pozadina svetla, može se dodati veća promena, dok HVS ne primeti razliku, nego u slučaju kada je pozadina relativno tamna.

Slika 6.5: Slika 'Fishingboat'. (a) Originalna slika. (b) Slika ravnomerno oštećena aditivnim belim Gausovim šumom

Slika 'Fishingboat' (slika 6.5) ravnomerno je oštećena aditivnim belim Gausovim šumom (AWGN). Šum je vidljiviji u tamnim delovima slike nego u svetlim, ako se uporede, na primer, tamni i svetli deo oblaka. Svetle oblasti mogu da prime više šuma dok ne postane primetan. Ovo svojstvo je našlo primenu u ugradnji digitalnog vodenog žiga.

Weber – Fechner-ov zakon datira iz sredine 19. veka. Kasnije je otkriveno da ovo tvrđenje nije baš tačno. Istraživanja su pokazala da prag opažanja raste sporije nego što je ovim zakonom dato. Danas postoje bolje funkcije promene praga opažanja.

Više promene je moguće uneti neprimetno u teksturisane, nego u ravne oblasti. Prag razlikovanja raste sa porastom detalja na slici. Aditivni slučajan šum je manje primetan u aktivnoj nego u jednostavnoj oblasti slike. Ljudsko oko je osetljivije u ravnim nego u teksturnim oblastima, gde intenzitet jako varira.

Prag razlikovanja raste sa porastom frekvence. Ovo svojstvo je nezavisno od slike. HVS je manje osetljiv na visokofrekventni sadržaj. Zato se u domenu transformacije mogu odbaciti neki visokofrekventni koeficijenti sa malim magnitudama, i na taj način postići kompresiju podataka bez primetnog uticaja na opažanje HVS-a.

7. Robusni algoritmi digitalnog vodenog žiga

Robusnost, kao možda najvažnije svojstvo digitalnog vodenog žiga, široko je izučavana tema. Do sada je predloženo mnogo metoda za robusne žigove za slike u nijansama sive.

7.1 Domen ugradnje

Klasifikacija algoritama digitalnog vodenog žiga obično zasniva na **domenu u kome se žig ugrađuje**. Tako, tehnike ugradnje su predložene

- u prostornom domenu ([7_01] – [7_03])
- u domenu transformacije, pre svega u DCT i blok DCT ([7_04] – [7_16]), Furijeovom ([7_17] – [7_19]) i domenu talasića ([7_20] – [7_23]).

Postoje i "hibridne metode", kod kojih se žig ugrađuje u dva različita domena (na primer, u prostornom i u domenu transformacije ([7_24])).

Ugradnja u prostornom domenu obično je dosta jednostavnija od one u domenu transformacije. Prostorni domen je "prirodni domen" za slike. Slika se javnosti predstavlja uvek u ovom obliku. U domenu transformacije ona je jedino dok je (i ako je) komprimovana na disku.

Ugradnja u **prostornom domenu** u suštini zahteva samo određenu promenu podataka slike – najčešće se to može shvatiti kao sabiranje matrice slike sa nekom drugom matricom.

Da bi se ugradnja obavila u **domenu transformacije**, potrebno je

1. prevesti sliku u domen transformacije
2. ugraditi poruku
3. vratiti sliku u prostorni domen

Da bi se takva poruka **detektovala**,

1. slika se prevede domen transformacije,
2. tu se detektuje poruka

Ugradnja informacije u domenu transformacije može biti daleko robusnija od ugradnje u prostornom domenu. Takvi metodi sakrivaju poruke u značajnim oblastima originalne slike, što ih čini daleko otpornijim na napade uklanjanja.

Ipak, treba primetiti da je takva ugradnja zaista robusna pre svega za **određenu vrstu** kompresije – onu koja se bazira na domenu u kome se ugradnja odvija. Na primer, žig ugrađen u blok DCT domenu pokazuje dobru robusnost prema JPEG, ali ne i prema kompresiji talasićima.

7.2 Tehnika raširenog spektra

U praksi žiga, veliku primenu ima *tehnika raširenog spektra*, prvobitno razvijena za vojne potrebe. Tajne vojne poruke su slate tako da su se širile kroz znatno veći raspon frekvencija nego što odgovara originalnom signalu; na celom opsegu na kome su se prostirale, imale su vrlo mali intenzitet. Na taj način su ostajale neprimećene za neprijatelja, jer su ostavljale samo utisak slabog pozadinskog šuma.

Širenje signala je određeno ključem koga primalac mora posedovati da bi detektovao signal. Komunikaciju raširenog spektra je vrlo teško ometati ili uklanjati, a verovatnoća neautorizovane detekcije je ekstremno mala. Ovde neprijatelj nije u stanju da pošalje signal odgovarajuće snage preko tako širokog spektra – ometanje nije praktično. Slično, u praksi žiga, neprijatelj ne može da doda odgovarajući šum da eliminiše žig, a da ne upropasti rad.

Tehnika raširenog spektra je najčešći način ugradnje žiga u digitalni rad. Ovakva ugradnja poseduje dobre kvalitete, u smislu održavanja vernosti rada, uz robusnost na veliki broj uobičajenih izobličenja, i uz sigurnost na većinu do sada korišćenih napada.

Veliki broj algoritama ugradnje u slike zasniva se na dodavanju neke matrice, koja odgovara sadržaju poruke, matrici slike. Ovde bit informacije ugrađujemo preko cele slike ili njenog velikog dela. To se realizuje tako što se matrica koja odgovara bitu poruke dodaje matrici slike. Poruka zbog tako postignute visoke redundantnosti može da odoli i dosta snažnim intervencijama koje se (sa ili bez namere uklanjanja žiga) obavljaju nad slikom posle ugradnje.

Ugradnja tehnikom raširenog spektra može se obaviti bilo u prostornom, bilo u nekom od domena transformacije. Najčešće, matrica koja sadrži poruku ima za elemente pseudoslučajne brojeve, male po apsolutnoj vrednosti (slici se dodaje šum). Takvih tehnika do sada je razvijeno mnogo ([7_03], [7_05] – [7_07], [7_17], [7_28] – [7_31]).

Među ovim algoritmima značajnu klasu čine oni kod kojih se u sliku ugrađuje beli Gausov šum (AWGN). Jednoj takvoj tehnici, Cox i dr. u [4_01] su posvetili dosta prostora (pre svega su se skoncentrisali na ugradnju u prostornom domenu). U daljem tekstu će ovi žigovi biti zvani *žigovi belog Gausovog šuma*.
Postoji i drugačija klasa algoritama, koja se takođe može svrstati u tehnike raširenog spektra. Bit poruke se ugradi u jedan ili nekoliko koeficijenata u domenu transformacije. Po povratku slike u prostorni domen, ugrađeni podaci se šire preko cele slike (ili preko bloka, u slučaju ugradnje u blok u domenu transformacije). To čini žig otpornim prema mnogim (posebno lokalnim) oštećenjima.

7.3 Informisana i slepa detekcija i ugradnja

Postoji i klasifikacija algoritama žiga zasnovana na **potrebi raspoloživosti originalne slike u vreme detekcije**:

- Algoritmi sa *informisanom detekcijom* zahtevaju prisustvo originalne slike u detektoru.
- U algoritmima sa *slepom detekcijom*, detektori ne iziskuju originalnu sliku.

Algoritmi sa informisanom detekcijom ([7_05] – [7_07]) u današnje vreme se retko koriste, i kadgod je to moguće (u zavisnosti od aplikacije žiga), zamenjuju se slepom ([7_01], [7_02], [7_14], [7_16], [7_23], [7_25] – [7_27]). U nekim aplikacijama, prisustvo originalne slike u detektoru je posebno nepoželjno. To se pre svega odnosi na žigove koji se koriste pri dokazivanju vlasništva u sudskom sporu. Informisana detekcija ovde obično dovodi do invertibilnosti algoritma, mogućnosti reverznog inženjeringa i čuvenog *Krejverovog napada* (videti potpoglavlje 6.5.2 *Neautorizovana ugradnja*).

Algoritmi ugradnje takođe mogu biti *slepi* i *informisani*. Za razliku od slepih, informisani algoritmi pre ugradnje ispitaju okolnosti – vezu između digitalne slike i poruke koja se ugrađuje, i u skladu sa tim okolnostima po potrebi koriguju obrazac koji se dodaje slici. Na taj način mogu se učiniti znatne uštede u potrebnoj snazi ugradnje.
Mnogi savršeniji algoritmi informisane ugradnje razvijeni su na bazi čuvenog rada Koste (Max Costa) "*Writing on Dirty Paper*" ([7_33]).
Kosta polazi od sledeće situacije iz svakodnevnog života.

Zamislimo list papira prekriven nezavisnim trunkama prljavštine sa normalno raspodeljenim intenzitetom. Pisac zna lokaciju i intenzitet komadića prljavštine, ali čitalac ne može da napravi razliku između njih i tragova mastila koje je ostavio pisac. Kosta razmatra problem pisanja poruke po takvom papiru, tako da se potroši za to najmanja količina mastila, potrebna da kasnije čitalac može da pročita poruku. Kaže da je, pri pisanju simbola (na primer slova "A") na prljavom papiru, najbolje to učiniti prilagođavanjem tog simbola komadićima prljavštine koji su već prisutni na papiru (informisano pisanje).

U kontekstu prakse žiga, prljavi papir je digitalni rad u koji se žig ugrađuje – poznat je ugrađivaču (piscu), ali ne i detektoru (čitaocu). Ako postoji **više kodova za svaku poruku, može se odabrati onaj koji najviše odgovara radu u koji se ugrađuje**, i tako ukloniti ometanje stvoreno originalnim radom (prljavim papirom). Za tehniku ugradnje žiga koja se na taj način realizuje koristi se naziv *informisano kodiranje.*

7.4 Nekoliko primera algoritama digitalnog vodenog žiga

Koch i Zhao ([7_04]) autori su jednog ranog algoritma. Da bi digitalni vodeni žig učinili robusnim prema JPEG kompresiji, ugradili su ga u 8×8 blokove u DCT domenu. Ugradnja bita informacije u DCT blok se u ovom algoritmu obavlja na sledeći način.

Unapred se odrede pozicije (u_1, v_1) i (u_2, v_2) u DCT bloku (iste za sve blokove slike), koji će učestvovati u ugradnji. Biraju se pozicije koje odgovaraju srednjim frekvencijama. Pozicije visokih frekvencija (ispod sporedne dijagonale bloka) nisu pogodne za ugradnju, jer tu ugrađena poruka verovatno ne bi preživela kvantizaciju. Ne preporučuje se ni ugradnja u koeficijentima niskih frekvencija (u blizini DC elementa), jer bi tu ugradnja bila previše uočljiva.

Jedan blok kodira "1" ako za koeficijente na odgovarajućim pozicijama, $d(u_1, v_1)$ i $d(u_2, v_2)$ važi $|d(u_1, v_1|>|d(u_2, v_2)|$, a "0" ako $|d(u_1, v_1|<|d(u_2, v_2)|$. Ako u startu koeficijenti ne zadovoljavaju ovaj odnos, treba im zameniti vrednosti, tako da odnos važi. Da bi se posle kompresije ovaj odnos očuvao (tj. da bi žig bio robusno ugrađen), ponekad se ugradnja dodatno pojača tako što se razlika vrednosti pri ugradnji poveća (većem koeficijentu se doda još neko $x > 0$. Što je x veće, algoritam će biti robusniji prema JPEG kompresiji, ali po cenu kvaliteta slike.

Ovaj algoritam omogućuje da se ugradi onoliko bitova koliko slika ima blokova. Tako, u sliku veličine 256x256 bitova moguće je ugraditi 32x32=1024 bita informacije.

Cox i dr. objavili su nekoliko radova ([7_05, 7_06, 7_07]) u vezi sa ugradnjom u domenu transformacije (transformacija se primenjuje na celu sliku). Radovi u vezi sa ovim algoritmom često su citirani od različitih autora, i predstavljaju početak korišćenja tehnike raširenog spektra u praksi digitalnog vodenog žiga (videti i novinski članak [7_32]). Autori predlažu ugradnju u DCT domenu, mada napominju da je moguće da se koristi i neki drugi domen (na primer, Furijeove ili transformacije talasićima). Oni ugrađuju poruku u n koeficijenata sa najvećim magnitudama, tako što se svakom od tih koeficijenata dodaje pseudoslučajan broj x_i $(i = 1,2,...,n)$, izabran u skladu sa raspodelom $N(0,1)$, pomnožen nekim faktorom snage α_i.

Povratkom slike u prostorni domen, promena svakog DCT koeficijenta širi se preko cele slike. Autori tvrde da ovaj žig preživljava i najintenzivniju JPEG kompresiju. Mana ove tehnike je informisana detekcija, tj. zahtev da originalna slika (bez žiga) bude raspoloživa u detektoru.

Wang i Kuo [7_20] primećuju da ugradnja u DCT koeficijente stvara žig koji nije uvek robustan prema kompresiji talasićima. To pre svega važi za algoritme sa ugradnjom u 8×8 DCT blokove. Oni sugerišu ugradnju u izabrane koeficijente transformacije talasićima, da bi žig preživeo ovu kompresiju.

Miller i dr. u svojim novijim radovima [7_34 – 7_37] predstavljaju jednu efikasniju, ali složeniju tehniku ugradnje žiga. Oni ugrađuju n bitova poruke u sliku sa n 8×8 blokova. Pri tome koriste tehniku raširenog spektra, poboljšanu **informisanim kodiranjem**. To se realizuje **pomoću koda prljavog papira**, u kome je svaka poruka predstavljena velikim brojem mogućih obrazaca, i u zavisnosti od vrednosti korelacije slike sa njima, donosi se odluka koji među njima će biti korišćeni u ugradnji.

8. Žig belog Gausovog šuma

Grupa tehnika raširenog spektra zasniva se na sabiranju matrice slike sa nekom matricom koja sadrži poruku. Kasnije, u fazi detekcije, ispituje se prisustvo takve matrice u slici u kojoj se žig traži.

U ovom poglavlju uvodi se jedna klasa tehnika, zasnovana na ugradnji belog Gausovog šuma, sa detekcijom zasnovanom na korelaciji.

Algoritam ugradnje i detekcije korišćen u daljem radu, preuzet je iz [4_01]. Nastavak rada (poglavlja 9 – 13) je **originalni rezultat ove disertacije**. U poglavlju 9 određuje se optimalna snaga za efikasnu ugradnju. Poglavlja 10 – 12 bave se snagom ugradnje za žig koji će biti izložen očekivanoj kompresiji (sa gubicima). U poglavlju 13 razmatra se robusnost prema drugim uobičajenim (valumetrijskim i geometrijskim) izobličenjima slike ([8_01]).

8.1. Ugradnja (jedan bit informacije)

Sliku u nijansama sive, veličine $m \times n$ piksela možemo predstaviti matricom c_0 dimenzije $m \times n$ ili (što je u suštini isto) vektorom dimenzije $m \cdot n$.[26] Komponente matrice (odnosno vektora) slike su vrednosti piksela – celi brojevi iz skupa $\{0,1,2,...,255\}$.

U takvu sliku ugrađujemo bit poruke na sledeći način:

Polazeći od unapred zadatog *ključa žiga* (koristi se kao seme generatora pseudoslučajnih brojeva), generiše se referentni obrazac r_w. *Referentni obrazac* je vektor pseudoslučajnih brojeva dimenzije slike ($m \cdot n$), **sa koordinatama iz standardne normalne (Gausove) raspodele (tj. koordinate mu podležu raspodeli $N(0,1)$). Matematičko očekivanje njegove norme (intenziteta) je $\sqrt{m \cdot n}$.**

[26] U daljem tekstu će se termini *vektor slike*, *matrica slike* i *slika* koristiti kao sinonimi (onda kada to ne bude stvaralo zabunu).

Ako se ugrađuje binarna jedinica, referentni obrazac se dodaje slici; ako se ugrađuje binarna nula, on se oduzima od slike.

Referentni obrazac r_w predstavlja vektor fiksnog intenziteta; ovaj intenzitet obično nije odgovarajući (nekad je prejak, a nekad preslab, zavisno od situacije u kojoj se primenjuje). Zato se pre ugradnje poruke, referentni obrazac prvo pomnoži nekim unapred određenim *koeficijentom snage ugradnje* $\alpha > 0$. Rezultat $\alpha \cdot r_w$ naziva se *vektor poruke*.

Tako, ugradnja bita poruke (binarne jedinice ili nule) u sliku c_0 obavlja se na sledeći način:

$$\boldsymbol{c_{w1} = c_0 + \alpha \cdot r_w}, \qquad \boldsymbol{c_{w0} = c_0 - \alpha \cdot r_w} \tag{8.1}$$

gde su c_{w1} i c_{w0} vektori koji nastaju pri ugradnji u sliku binarne jedinice, odnosno nule.

Rezultujuće vrednosti u matricama se pri snimanju na disk "udevaju" u dopustive vrednosti piksela slike. Zato je konačni efekat ovih operacija

$$c_{w1} = [c_0 + \alpha \cdot r_w]_8, \qquad c_{w0} = [c_0 - \alpha \cdot r_w]_8 \tag{8.2}$$

Oznaka $[\ ]_8$ je za "udevanje" koordinata rezultujućeg vektora u 8-bitne vrednosti, tj. u vrednosti iz skupa $\{0,1,2,...,255\}$. [27]

8.2. Detekcija

Da bi odgovorio da li je u neku sliku (c) ugrađen žig, detektor polazi od ključa žiga (istog kao u ugrađivaču). Iz ključa se generiše referentni obrazac r_w (isti kao kod ugrađivača). Izlaz iz detektora je detektovana poruka žiga.

Provera da li je takav referentni obrazac ugrađen u sliku ili nije obavlja računanjem neke forme korelacije vektora slike $c = (c_1, c_2, ..., c_{mn})$ i referentnog obrasca $r_w = (r_{w1}, r_{w2}, ..., r_{wmn})$ – izračunata vrednost korelacije $z(c, r_w)$ poredi se zatim sa unapred određenim *pragom* τ. Detektor će izvestiti:

- poruka je 1 ako $z(c, r_w) > \tau$
- poruka je 0 ako $z(c, r_w) < -\tau$ (8.3)
- nema poruke ako $|z(c, r_w)| \le \tau$

[27] Ova primedba važi za sve rezultate operacija – matrica slike pri snimanju na disk prolazi uvek kroz ovaj proces "udevanja" u dopuštene vrednosti.

Najčešće korišćene mere korelacije su:

Linearna korelacija

$$lc(c, r_w) = \frac{c \bullet r_w}{\|r_w\| \cdot \|r_w\|} \tag{8.4}$$

($\bullet$ je oznaka skalarnog proizvoda, a $\| \ \|$ – norme vektora)

Normalizovana korelacija

$$nc(c, r_w) = \frac{c \bullet r_w}{\|c\| \cdot \|r_w\|} = \cos(c, r_w) \tag{8.5}$$

Korelacioni koeficijent

$$cc(c, r_w) = nc(\bar{c}, \overline{r_w}) \tag{8.6}$$

$$\bar{c} = c - c^{sr}, \qquad \overline{r_w} = r_w - r_w{}^{sr},$$

$$c^{sr} = \frac{(c_1 + c_2 + ... + c_{mn})}{mn} \cdot (1,1,...,1), \qquad r_w{}^{sr} = \frac{(r_{w1} + r_{w2} + ... + r_{wmn})}{mn} \cdot (1,1,...,1)$$

U algoritmu u daljem radu koristiće se **linearna korelacija**.

8.3. Duža poruka

Duža poruka (više od jednog bita informacije) može se u sliku ugraditi na različite načine.

Ugradnja bitova poruke jedan preko drugog: Celoj slici dodaje se vektor poruke dimenzije slike, koji odgovara prvom bitu; zatim se generiše (i dodaje slici) sledeći vektor poruke, koji odgovara sledećem bitu,...

Ugradnja u podslike: Slika se deli u disjunktne podslike, u koje se zatim upisuje po jedan bit poruke. *Podslika* je proizvoljan podskup piksela slike (ona može, ali ne mora, biti sastavljena od susednih piksela). Detektor će sliku podeliti na iste podslike kao ugrađivač, i u svakoj od njih tražiti bit poruke.

Moguće su različite varijante navedena dva metoda. Na primer, moguće je u svaku podsliku uneti nekoliko vektora poruke (svaki za po jedan bit poruke).

8.4 Robustan algoritam i robustan žig

Uslov robusnosti nije binarni: jedna tehnika je manje ili više robusna od druge. S druge strane, tehnika može da bude robusnija od druge u odnosu na neku klasu napada, a manje robusna u odnosu na neku drugu. Uobičajeno je da se robusnim smatraju one

tehnike čiji žigovi su u stanju da odole onim operacijama – napadima uklanjanja koji zadržavaju dobar nivo vernosti.

Svi u prethodnom poglavlju spomenuti algoritmi su robusni. Ni algoritam žiga belog Gausovog šuma nije izuzetak – on pokazuje dobre karakteristike robusnosti prema mnogim napadima uklanjanja.

Jasno, **nije svaki poseban žig ugrađen robusnom tehnikom – robustan:** takav žig **mora biti ugrađen dovoljno snažno**. Međutim, prejako ugrađen žig uništiće kvalitet slike. Zato je važno naći pravu meru za snagu ugradnje.

To je tema ovog rada: ovde se ne predstavlja novi algoritam žiga, nego se za dobro poznat, robustan algoritam, određuje *optimalna* snaga ugradnje –minimalna, koja sa verovatnoćom od praktično 100% osigurava detektabilnost posle očekivanog napada.

Ovaj rad nastao je kao rezultat traganja za merom snage ugradnje žiga koji bi bio **robustan prema očekivanoj kompresiji**. Precizno, koliko žig treba snažno ugraditi u sliku koja će zatim biti izložena poznatoj (na pr. DjVu ili JPEG) kompresiji, da bi ostao detektabilan u komprimovanom fajlu. To je razlog što je najveći deo teksta posvećen napadu kompresije. Samo se poglavlje 13 (*Žig belog Gausovog šuma i druge modifikacije slike*) bavi ugradnjom, robusnom prema drugim napadima.

8.5. Beli Gausov šum

Šum se definiše kao slučajan ili ponavljajući događaj koji zamagljuje ili ometa korisnu informaciju. Jasno, šum se može predstaviti vektorom.

Vektor šuma je *beo* ako su vrednosti njegovih koordinata nekorelirane i imaju jediničnu varijansu. Beli šum mora imati nula autokorelaciju sa sobom u prostoru, osim za pomeranje (šiftovanje) jednako nuli. Obrnuto, ako autokorelacija signala ima ta svojstva (0 osim za 0 šift), signal je beo.

To što je signal nekoreliran sa svojom šiftovanom kopijom ne ograničava vrednosti koje on može da ima. Svaka raspodela vrednosti je moguća. Na primer, binarni signal koji uzima samo vrednosti +1 i -1 biće beo ako je sekvenca ovih vrednosti statistički nekorelirana. Šum koji ima neprekidnu raspodelu, kao što je normalna raspodela, takođe može biti beo.

Često se pogrešno misli da je *Gausov šum* (šum sa normalnom raspodelom) neophodno beli šum. Dva svojstva (beo i Gausov) ne povlače jedno drugo. Beli Gausov šum je dobra aproksimacija za mnoge situacije iz realnog sveta.

Aditivan šum je šum koji se dodaje radu.

Aditivan beli Gausov šum (AWGN) je aditivan šum koji je istovremeno beo i Gausov. Ovde uvedeni referentni obrazac predstavlja obrazac belog Gausovog šuma. U praksi se generiše pomoću *Matlab* funkcije *randn* (svaki element se dobija kao izlaz generatora pseudoslučajnih brojeva, sa standardnom normalnom raspodelom).

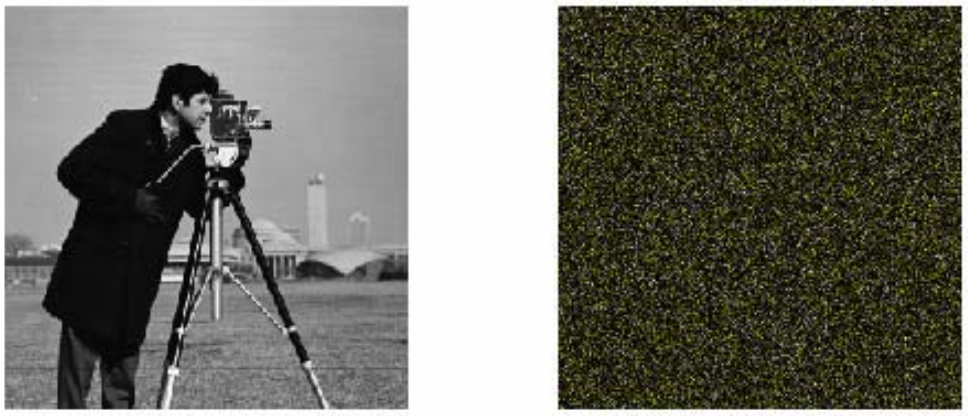
8.6. Geometrijska interpretacija slike i referentnog obrasca

Slika dimenzija $m \times n$, čije vrednosti piksela pripadaju skupu $\{0,1,...,255\}$, može se posmatrati kao tačka (vektor) u realnom vektorskom prostoru dimenzije $m \cdot n$. Međutim, koordinate tih vektora uzimaju samo celobrojne vrednosti od 0 do 255. Rezultati operacija nad vektorima, ukoliko nisu celobrojne vrednosti iz ovog skupa, po izvršenim operacijama "udevaju" se u ove vrednosti. Tako, mogućih slika dimenzije $m \times n$ u nijansama sive ima ogroman, ali ipak konačan broj: 256^{mn}.

Najveći broj vektora – tačaka iz ovog skupa ne predstavlja nikakvu "prirodnu" sliku. Ove slike obično imaju veliku prostornu redundancu, jer susedni pikseli u najvećem broju slučajeva imaju vrlo bliske vrednosti.

Referentni obrasci takođe su vektori prostora R^{mn}. Njihove vrednosti koordinata nisu brojevi iz skupa $\{0,1,...,255\}$, nego iz raspodele $N(0,1)$ (pozitivni i negativni realni brojevi, uglavnom mali po apsolutnoj vrednosti).

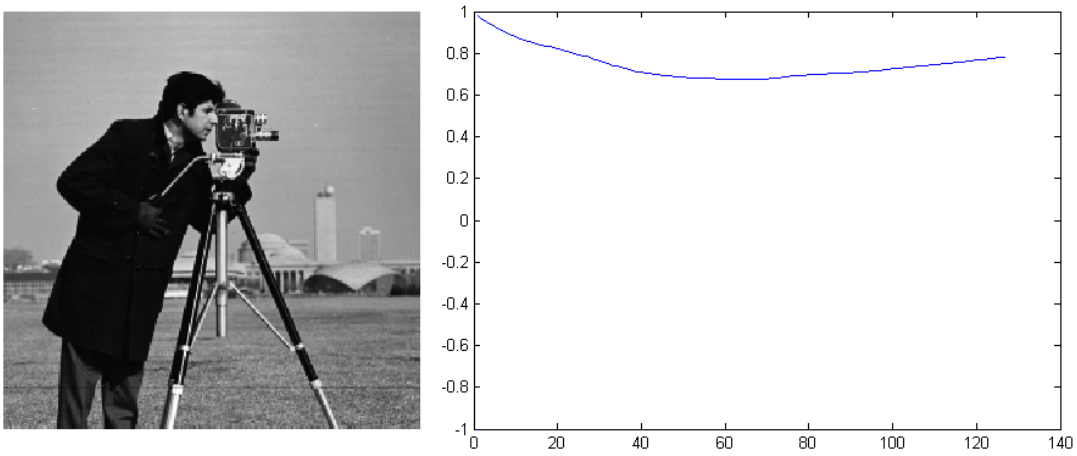
Na slici 8.1 predstavljeni su primeri slike i referentnog obrasca.



Slika 8.1: Primeri originalne slike i referentnog obrasca

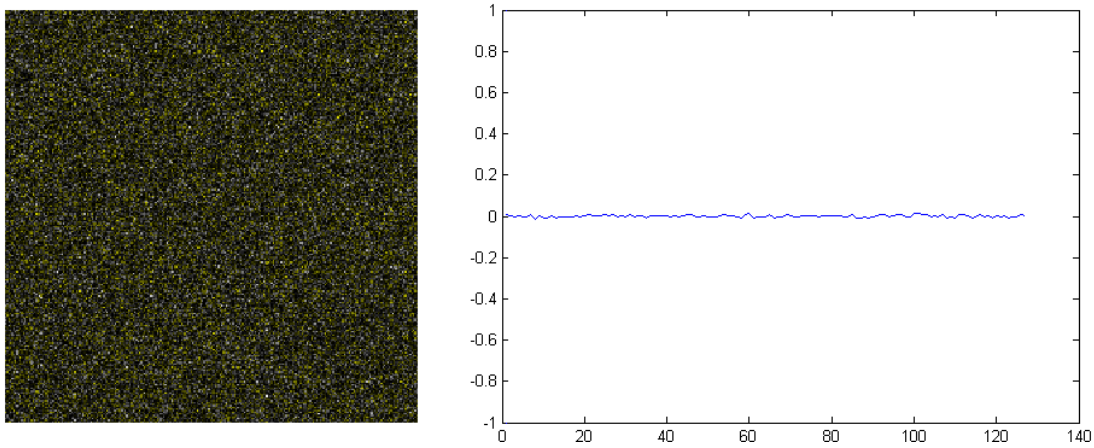
Koliko je matrica referentnog obrasca različita po svojoj prirodi od matrice slike pokazuje sledeći test, kojim se dobija grafik autokorelacije za sliku i za referentni obrazac, šiftovano od jednog piksela, pa do polovine širine matrice (slike 8.2 i 8.3). Vidi se da je slika visoko korelirana sa sopstvenom šiftovanom verzijom, za razliku od referentnog obrasca, koji je sa svojim šiftom praktično nekoreliran.

Ovde je korišćena mera normalizovane korelacije, kao očiglednija – normalizovana korelacija dva vektora jednaka je *kosinusu* ugla među njima. Ako je jednaka $+1$ ili -1, vektori su kolinearni (pa zato i visoko korelirani). Ako je 0, oni su među sobom ortogonalni (pa time i nekorelirani).



Slika 8.2: Ilustracija koreliranosti slike sa sopstvenom šiftovanom verzijom, za sliku 'Cameraman' (dimenzije 256×256). Apscisa predstavlja šift originalne slike; ordinata prikazuje vrednost normalizovane korelacije

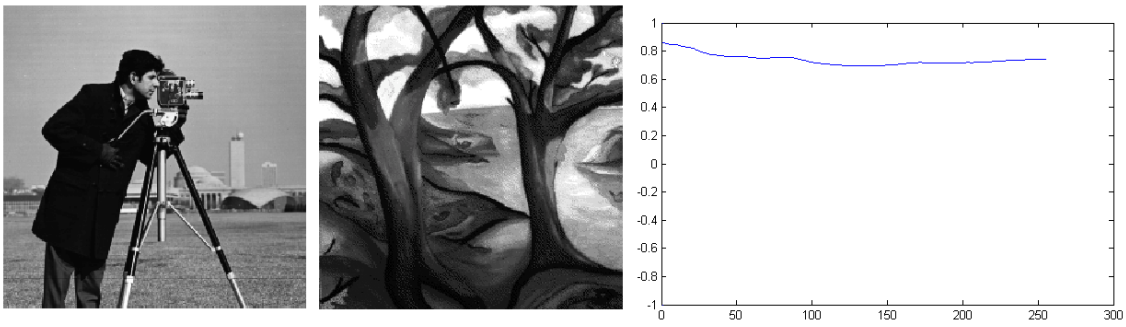
... i isto to za referentni obrazac:



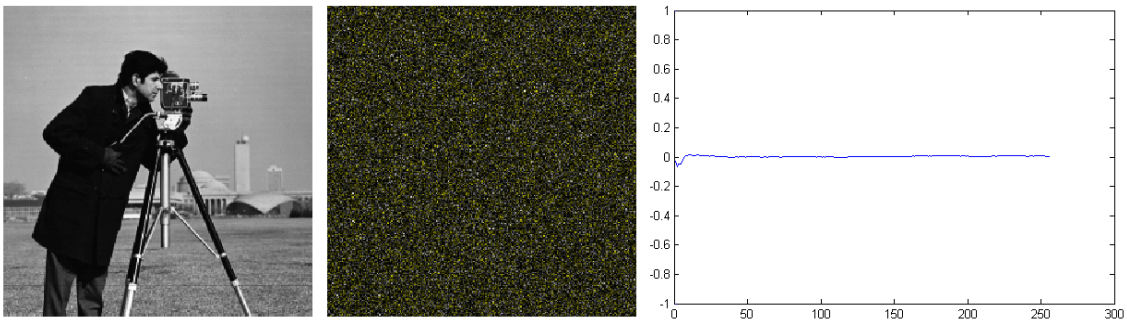
Slika 8.3: Ilustracija koreliranosti vektora belog Gausovog šuma sa svojom šiftovanom verzijom

Može se videti da je razlika korelacije za sliku i za referentni obrazac sa sopstvenim šifrovanim verzijama velika. Za sliku, čak i za veliki šift nije manja od 0.7, dok je za referentni obrazac u navedenom slučaju ne veća od 0.015.

Na slikama 8.4 i 8.5 prikazane su vrednosti korelacije između dve slike, i između slike i referentnog obrasca.



Slika 8.4: Korelacija dve različite slike (dimenzije slika su 256×256); na grafiku je prikazana vrednost normalizovane korelacije za prvih 1, 2, ..., 256 kolona matrica slika



Slika 8.5: Korelacija slike i referentnog obrasca (dimenzije su 256×256); na grafiku je prikazana vrednost normalizovane korelacije za prvih 1, 2, ..., 256 kolona matrica

Na osnovu prethodnih slika može se zaključiti:

- Čak i različite slike su uglavnom među sobom visoko korelirane. Sličan rezultat vrednosti korelacije dobija se za sliku i njenu šiftovanu verziju. Jedino za vrlo mali šift (ne veći od 10) korelacija slike sa sopstvenim šiftom je vrlo visoka (preko 0.9). Veći šift dovodi do korelacije približno iste sa vrednošću za različite slike.
- Referentni obrazac je praktično potpuno nekoreliran i sa sopstvenim šiftom, i sa proizvoljnom slikom (takođe i sa drugim referentnim obrascem) – vrednosti normalizovane korelacije sasvim malo odstupaju od nule.

3. deo: Optimalna snaga ugradnje

U ovom delu dati su **originalni rezultati** nastali u pokušaju da se odredi optimalna snaga ugradnje.

9. Određivanje snage za efikasnu ugradnju

Žig je *ugrađen efikasno* ako je detektabilan neposredno po ugradnji.

Ako se poruka može detektovati u slici koja je posle ugradnje bila izložena nekom napadu uklanjanja, kaže se da je žig *robustan prema pretrpljenom napadu*.

U potpoglavlju 9.1 određuje se minimalna vrednost koeficijenta snage (α) za efikasnu ugradnju jednog bita poruke, a u potpoglavlju 9.2 – za dužu poruku.

9.1. Efikasna ugradnja jednog bita poruke

9.1.1. Važnost dobrog izbora α i τ

Koeficijent snage ugradnje α i prag detekcije τ direktno utiču na efikasnost ugradnje.

Ukoliko se žig ugradi snažno (sa velikim α), moći će se dostići ideal 100% efikasne ugradnje. Međutim, to može da loše utiče na kvalitet slike, i da promene na njoj nastale ugradnjom žiga postanu primetne.

Jasno, ako se smanji prag detekcije τ (postavi se da bude blizak, ili čak jednak nuli), ugradnja će automatski biti efikasnija (detektor će u znatno većem procentu slučajeva ugradnje detektovati žig u slici). Zašto onda uopšte imati prag τ? Ako je $\tau = 0$, detektor nikada neće izvestiti da poruke nema. Izvestiće uvek da je žig ugrađen – čak i kada nije. Tako, ako je prag τ premali, verovatnoća lažnog pozitivnog će porasti.

Vrlo je važno pronaći meru – odrediti koeficijent α i prag τ, tako da verovatnoće lažnog negativnog (neefikasna ugradnja) i lažnog pozitivnog (slučaj kada detektor odgovara da je žig ugrađen u sliku kada to nije tačno) budu prihvatljivo male. Takođe i da žig ne bude ugrađen prejako, da se ne bi ugrozila vernost.

9.1.2. Odstupanje $lc(c_0, r)$ od nule – parametar σ_{lc}

U pronalaženju takve mere *optimalne ugradnje*, polazi se od poznatih činjenica za normalnu raspodelu:

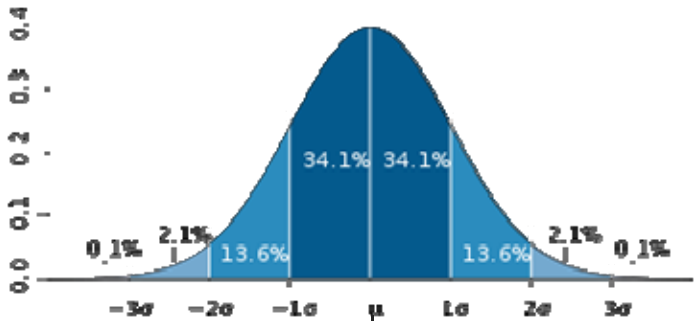
- Ako su $X_1 \sim N(\mu_1, {\sigma_1}^2)$, $X_2 \sim N(\mu_2, {\sigma_2}^2)$, onda će njihova linearna kombinacija takođe biti normalno raspodeljena:

$$aX_1 + bX_2 \sim N(a\mu_1 + b\mu_2, a^2{\sigma_1}^2 + b_2{\sigma_2}^2) \tag{9.1}$$

- Funkcija gustine verovatnoće za normalnu raspodelu $N(\mu, \sigma^2)$ data je formulom

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)} = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right) \tag{9.2}$$

i prikazana na slici



- Za podatke iz raspodele $N(\mu, \sigma^2)$ važi *Pravilo 68–95–99,7 (empirijsko pravilo)*:
  - oko 68% vrednosti nalazi se u intervalu $(\mu - \sigma, \mu + \sigma)$,
  - oko 95% vrednosti nalazi se u intervalu $(\mu - 2\sigma, \mu + 2\sigma)$, (9.3)
  - oko 99,7% vrednosti nalazi se u intervalu $(\mu - 3\sigma, \mu + 3\sigma)$
- Suma kvadrata k standardnih normalnih slučajnih promenljivih je χ^2 (Hi-kvadrat) promenljiva sa k stepeni slobode. Matematičko očekivanje joj je k.

U tekstu koji sledi, za slike i referentne obrasce biće korišćene sledeće oznake:

c_0 – originalna slika (odnosno slika u koju referentni obrazac r_w nije ugrađen)

c – slika koja je ulaz u detektor (u nju možda jeste, a možda i nije ugrađen obrazac r_w)

r_w – referentni obrazac koji se ugrađuje

r – proizvoljni referentni obrazac

Sve koordinate proizvoljnog referentnog obrasca $r=(r(1),r(2),...,r(mn))$ podležu raspodeli $N(0,1)$. Matematičko očekivanje za $\|r\|$ je $\sqrt{mn}$. Za sliku $c_0=(c_0(1),c_0(2),...,c_0(mn))$ i referentni obrazac r, linearna korelacija je

$$lc(c_0,r)=\frac{c_0 \bullet r}{\|r\|^2}=\frac{\sum_{i=1}^{mn} c_0(i)r(i)}{mn} \tag{9.4}$$

Zato $lc(c_0,r)$, kao linearna kombinacija slučajnih promenljivih $r(1),r(2),...,r(mn)$ uzima vrednost iz normalne raspodele $N(0,\sigma_{lc}^2)$, gde je:

$$\sigma_{lc}=\frac{\sqrt{(c_0(1))^2+(c_0(2))^2+...+(c_0(mn))^2}}{mn}=\frac{\sqrt{E(c_0)}}{mn} \tag{9.5}$$

Važi i:

- u intervalu $(-\sigma_{lc},\sigma_{lc})$ nalazi se 68% vrednosti linearne korelacije,
- u intervalu $(-2\sigma_{lc},2\sigma_{lc})$ nalazi se 95% vrednosti linearne korelacije,
- u intervalu $(-3\sigma_{lc},3\sigma_{lc})$ nalaze se gotovo sve vrednosti linearne korelacije.

9.1.3. Određivanje koeficijenta snage ugradnje α

Na osnovu empirijskog pravila, sa verovatnoćom 0.997 ("skoro 1") je $|lc(c_0,r_w)|\le 3\sigma_{lc}$

Parametar σ_{lc} (standardna devijacija uzorka linearnih korelacija originalne slike sa referentnim obrascima) predstavlja osnovu za pravilno određivanje parametara α i τ, za efikasnu ugradnju.

Potrebna snaga ugradnje najveća je

- kada $lc(c_0,r_w)=-3\sigma_{lc}$, a ugrađuje se binarna jedinica i
- kada $lc(c_0,r_w)=3\sigma_{lc}$, a ugrađuje se binarna nula

Zato, ako se uzme

$$\alpha=3\sigma_{lc}+\tau, \tag{9.6}$$

sa vrlo velikom sigurnošću će žig biti efikasno ugrađen.

Takva ugradnja kod koje se α unapred odredi iz parametra σ_{lc}, pa obrazac ugradi u sliku tom snagom (ne uzima se u obzir koja je vrednost linearne korelacije slike sa referentnim obrascem **koji se ugrađuje**), naziva se *ugradnja fiksnom snagom*.

Fiksna snaga ugradnje ima veliku manu: obično je dosta veća nego što je to neophodno. Zato se često u određivanju vrednosti α, osim σ_{lc} uzima u obzir i vrednost linearne korelacije slike sa obrascem **koji se ugrađuje**. U ovom slučaju se kaže da se u algoritmu *podešava snaga ugradnje*.

Neka je $lc(c_0, r_w) = l$. Ako $l < \tau$ i ugrađuje se binarna jedinica, treba uzeti $\alpha = \tau - l$. Ako $l \geq \tau$, biće $\alpha = 0$ (nije potrebno da se bilo šta ugrađuje). Znači,

- ako se ugrađuje binarna jedinica, bira se $\alpha = \max(\tau - l, 0)$ (9.7')
- ako se ugrađuje binarna nula, bira se $\alpha = \max(\tau + l, 0)$. (9.7")

Ugradnja snagom α čini na slici srednju kvadratnu grešku

$$MSE(c_0, c_w) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[c_w(i,j) - c_0(i,j)\right]^2 = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha^2\left[r_w(i,j)\right]^2 = \alpha^2 \qquad (9.8)$$

9.1.4. Izbor praga detekcije τ

Da bi se izbegla greška lažnog pozitivnog, τ treba da bude veće od najveće vrednosti korelacije koju slika može da ima sa referentnim obrascem koji u nju nije ugrađen.

Ako se uzme $\tau = 3\sigma_{lc}$, vrednost linearne korelacije originalne slike (c_0) sa referentnim obrascem biće skoro sa verovatnoćom $p = 1$, u intervalu $(-\tau, \tau)$. Drugim rečima, biće skoro 100% sigurno da detektor neće načiniti grešku lažnog pozitivnog (tj. izvestiti da je žig ugrađen onda kada to nije slučaj).

Prag ne treba da bude veći od $3\sigma_{lc}$: to ne samo što je nepotrebno, nego i negativno utiče na vernost (biće potrebna snažnija ugradnja, da bi detektor prepoznao poruku).

Greška lažnog pozitivnog je nešto o čemu treba voditi računa u slučaju sasvim kratke poruke (kao ovde, u slučaju poruke od jednog bita). Ako je poruka duža, ova greška i nije tako veliki problem (o tome će biti više reči kasnije), i za τ se može uzeti i znatno manja vrednost.

9.1.5. Zavisnost parametra σ_{lc} od dimenzije slike

Parametar σ_{lc} je manji za veće slike. Tako, slika c_k nastala iz k jednakih slika, c, energije $E(c)$, imaće energiju $E(c_k) = k \cdot E(c)$ i dimenziju $\dim(c_k) = k \cdot \dim(c)$. Njen σ_{lc} parametar će biti

$$\sigma_{lc}(c_k) = \frac{\sqrt{E(c_k)}}{\dim(c_k)} = \frac{\sqrt{k \cdot E(c)}}{k \cdot \dim(c)} = \frac{\sqrt{E(c)}}{\sqrt{k} \cdot \dim(c)} = \frac{1}{\sqrt{k}} \cdot \sigma_{lc}(c) \qquad (9.9)$$

Zato, u slici sa većim dimenzijama, poruka se ugrađuje manjom snagom. U prethodnom slučaju, biće

$$\alpha_k = \alpha/\sqrt{k} \text{ i } \tau_k = \tau/\sqrt{k} \tag{9.10}$$

Ako se vodi računa o vernosti slike, jasno je da se u veću sliku može ugraditi duža poruka.

Parametri σ_{lc}, α i τ se povećavaju sa porastom energije. Ako su c_1 i c_2 slike jednakih dimenzija, i $E(c_1) > E(c_2)$, tada se u sliku c_2 može ugraditi duža poruka. Tako, **duža se poruka može ugraditi u tamniju nego u svetliju sliku**[28].

9.2. Efikasna ugradnja duže poruke

9.2.1. Izbor praga τ kod duže poruke

Prag τ kod ugradnje duže poruke može da dobije i dosta manju vrednost od $3\sigma_{lc}$ (vrednost preporučena u slučaju poruke od jednog bita). Na primer, za $\tau = \sigma_{lc}$, za približno 68% mogućih referentnih obrazaca linearna korelacija slike sa referentnim obrascem biće unutar intervala $(-\tau, \tau)$. To znači da se lažno pozitivno može pojaviti u 'svakom trećem' slučaju. Ako se zahteva da se svaki bit poruke mora detektovati da bi se prisustvo poruke potvrdilo, biće gotovo nemoguće da se poruka detektuje ako nije ugrađena (za tako nešto bilo bi potrebno da se za sve referentne obrasce pojavi lažno pozitivno, što je kod duže poruke gotovo nemoguće).

Tako je u slučaju duže poruke moguće koristiti i dosta manju vrednost τ.

9.2.2. Ugradnja jedne poruke preko druge

Zahvaljujući činjenici da su referentni obrasci među sobom gotovo sasvim nekorelirani, ugradnja novog obrasca neće bitno promeniti linearnu korelaciju slike sa prethodno ugrađenim obrascima:

$$lc(c_0 + r_1 + r_2, r_1) = \frac{(c_0 + r_1) \bullet r_1 + r_2 \bullet r_1}{\|r_1\| \cdot \|r_1\|} \approx \frac{(c_0 + r_1) \bullet r_1 + 0}{\|r_1\| \cdot \|r_1\|} = lc(c_0 + r_1, r_1) \tag{9.11}$$

Drugim rečima, **linearna korelacija je otporna na beli Gausov šum.**

Zato, kada se ugrađuju bitovi poruke jedan preko drugog, može se za α uzeti praktično ista vrednost kao u slučaju ugradnje jednog bita informacije.[29]

---

[28] Savet da žig zato treba ugrađivati u tamnije oblasti slike baš i nije primenljiv, jer s druge strane, **ljudsko oko je manje osetljivo na promene sjajnosti u svetlijim nego u tamnijim oblastima slike**.

U sliku $c_0 = (c_0(1), c_0(2), ...c_0(mn))$ ugrađuje se k bitova tako što se za svaki bit dodaje slici (ili oduzima od nje) odgovarajući referentni obrazac $r_j = (r_j(1), r_j(2), ..., r_j(mn))$ ($j = 1,2,...,k$), pomnožen snagom ugradnje α_j. Rezultujuća slika je $c_w = (c_w(1), c_w(2), ...c_w(mn))$. Svaki piksel slike dobija se sa

$$c_w(i) = c_0(i) + \sum_{j=1}^{k} (\pm\alpha_j \cdot r_j(i)) \qquad (i = 1,2,...,mn) \tag{9.12}$$

Ovakva ugradnja je lokalizovana u prostoru. Samo odgovarajuće koordinate originalne slike i referentnog obrasca utiču na vrednost piksela rezultujuće slike.

Koordinate referentnih obrazaca uzimaju vrednosti iz normalne raspodele $N(0,1)$. Zato, njihova linearna kombinacija

$$R(i) = \sum_{j=1}^{k} (\pm\alpha_j \cdot r_j(i)) \quad (i = 1,2,...,mn) \tag{9.13}$$

uzima vrednosti iz $N(0,\sigma^2)$, gde je

$$\sigma = \sqrt{\alpha_1^2 + \alpha_2^2 + ... + \alpha_k^2} \tag{9.14}$$

Takođe, koordinate vektora $r_s = (r_s(1), r_s(2), ..., r_s(mn))$, gde je

$$r_s(i) = \frac{R(i)}{\sqrt{\alpha_1^2 + \alpha_2^2 + ... + \alpha_k^2}}, \ (i = 1,2,...,mn) \tag{9.15}$$

podležu raspodeli $N(0,1)$, pa je r_s referentni obrazac.

Tako, ugradnja k referentnih obrazaca r_j jednog preko drugog, snagama α_j ($j = 1,2,...,k$), jednaka je ugradnji jednog referentnog obrasca, r_s, snagom $\beta = \sqrt{\alpha_1^2 + \alpha_2^2 + ... + \alpha_k^2}$, ili (9.16)

$$c_w = c_0 + \beta \cdot r_s. \tag{9.17}$$

Ako su svi obrasci ugrađeni istom snagom, $\alpha_1 = \alpha_2 = ... = \alpha_k = \alpha$, tada je ukupna snaga ugradnje

$$\beta = \sqrt{\alpha_1^2 + \alpha_2^2 + ... + \alpha_k^2} = \sqrt{k \cdot \alpha^2} = \sqrt{k} \cdot \alpha \tag{9.18}$$

Primer 9.1: Ako $k = 2$, $\alpha_1 = 3$, $\alpha_2 = 4$, tada

---

[29] (Ako se zanemari činjenica da je kod ugradnje jednog bita informacije do sada uzimano $\tau = 3 \cdot \sigma_{lc}$, i pretpostavi da se i tu za τ može uzeti i manja vrednost).

$$r_s = \frac{3 \cdot r_1 + 4 \cdot r_2}{5}$$

Efekat ugradnje dva obrasca snagama 3 i 4 ekvivalentan je ugradnji jednog obrasca, snagom $\beta = \sqrt{\alpha_1{}^2 + \alpha_2{}^2} = \sqrt{3^2 + 4^2} = 5$.

Primer 9.2: Ako je $\alpha = 2$, i ugrađuje se poruka od $k = 25$ bitova (jedan preko drugog, fiksnom snagom ugradnje), biće $\beta = \sqrt{k} \cdot \alpha = 10$ (ugradnja 25 obrazaca snagom 2 je ekvivalentna ugradnji jednog obrasca snagom 10).

9.2.3. Ugradnja u podslike

Slika c_0 dimenzije mn piksela deli se na k disjunktnih podslika $\boldsymbol{c_0^1, c_0^2, ..., c_0^k}$, redom dimenzija $mn^1, mn^2, ..., mn^k$ ($mn = \sum_{j=1}^{k} mn^j$). Za svaku podsliku, koeficijent σ_{lc}^j ($j = 1,2,...,k$) je

$$\sigma_{lc}^j = \frac{\sqrt{E(c_0^j)}}{mn^j} \tag{9.19}$$

Ako je moguće sliku izdeliti u k delova jednakih dimenzija i energija, tada će sve podslike imati istu vrednost parametra σ_{lc}^j:

$$\sigma_{lc}^j = \frac{\sqrt{E(c_0^j)}}{mn^j} = \frac{\sqrt{E(c_0)/k}}{mn/k} = \sqrt{k} \cdot \frac{\sqrt{E(c_0)}}{mn} = \sqrt{k} \cdot \sigma_{lc} \tag{9.20}$$

U tom slučaju, ukupna snaga za poruku od k bitova, koja se ugrađuje u k podslika, jednaka je snazi za jedan bit poruke, pomnoženoj sa $\sqrt{k}$. Snaga ugradnje u tom slučaju **jednaka je kao da je poruka ugrađena jedan bit preko drugog**.

10. Kompresija i robustan žig

U nekim situacijama može se očekivati da će slika sa žigom od trenutka ugradnje do detekcije biti izložena nekoj određenoj modifikaciji (kompresiji, opsecanju, promeni dimenzije, ...). U takvim okolnostima korisno je znati koliko snažno žig treba ugraditi, da bi i posle te modifikacije bio detektabilan.

Kompresija među ovakvim modifikacijama slike zauzima svakako vrlo važno mesto. Ne samo zato što se može očekivati da neprijatelj pokuša da je upotrebi za uklanjanje žiga. **Ugradnja žiga u JPEG ili DjVu sliku i ne može biti ugrađena drugačije, nego da se posle ugradnje slika izloži kompresiji!**

Kako su ovo fajlovi (videti kraj poglavlja 2) koje ne treba modifikovati, nego ih treba koristiti samo kao "konačan proizvod", "fotokopiju", žig se u njih ne ugrađuje. Ako hoćemo da stvorimo JPEG fajl sa žigom, postupak je sledeći:

1. U originalnu (na pr. TIFF sliku) ugradimo poruku žiga;
2. Sliku sa žigom sačuvamo kao JPEG fajl željenog kvaliteta.

U drugom koraku, poruka žiga će u izvesnoj meri biti uništena. Pitanje je zato koliko snažno u originalnu sliku ugraditi žig, da bi bio detektabilan u JPEG slici.

Tekst koji sledi daje procenu koeficijenta snage α, da bi se ugradio žig koji će preživeti očekivanu kompresiju.

U potpoglavlju 10.1 razmatra se slučaj jednog bita poruke, a u potpoglavlju 10.2 – slučaj duže poruke. U svrhu jednostavnijeg izlaganja, biće razmatran samo slučaj ugradnje binarne jedinice; sve ovde rečeno može se lako primeniti i na slučaj binarne nule.[30]

---

[30] Ograničavanje ovde (i bilo gde dalje) u tekstu na ugradnju binarne jedinice ne umanjuje opštost razmatranja. Na ugradnju binarne nule možemo gledati i kao na ugradnju suprotnog obrasca. Suprotni obrazac od r_w je $-r_w$, i takođe ima sva svojstva referentnog obrasca (podleže raspodeli $N(0,1)$).

10.1. Robusnost prema kompresiji – jedan bit poruke

10.1.1. Koeficijent snage ugradnje posle kompresije (α')

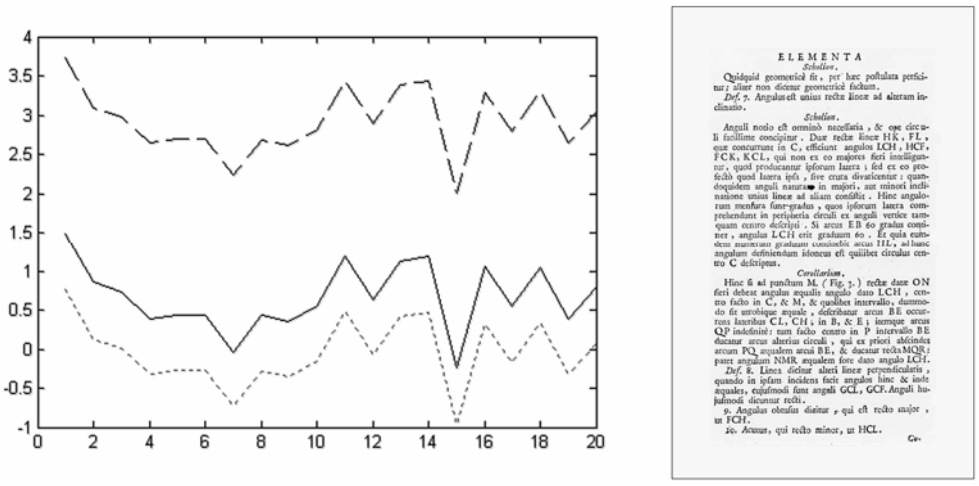
Cilj sledećeg eksperimenta je da se ustanovi koliki deo žiga će biti uništen kompresijom. Na taj način će se moći odrediti koeficijent snage ugradnje koji je dovoljan da obezbedi robusnost prema očekivanoj kompresiji.

U ovom eksperimentu ugrađuje se nekoliko (k) referentnih obrazaca $r(1), r(2), ..., r(k)$ (jedan po jedan) u sliku snagom α (broj α je isti za svih k obrazaca). Tako se dobija k slika sa žigom (ugrađenom binarnom jedinicom) $c_w(1), c_w(2), ..., c_w(k)$.

Zatim se svaka od tih k slika podvrgava očekivanoj kompresiji (istoj za svih k slika). Rezultat je k komprimovanih slika sa žigom $c_{wn}(1), c_{wn}(2), ..., c_{wn}(k)$.

Za svaku od tih k komprimovanih slika sa žigom računa se linearna korelacija sa odgovarajućim referentnim obrascem (onim koji je u nju prethodno ugrađen).

Sledeći primer (slika 10.1) prikazuje navedene vrednosti linearnih korelacija slika sa referentnim obrascima (c_0 je sken prve strane knjige 'Elementa geometriae' Ruđera Boškovića, $\alpha = 3$, $k = 20$, kompresija je DjVu Photo). Apscisa grafika predstavlja indekse nizova, a ordinata – njihove vrednosti[31].



Slika 10.1: Linearne korelacije slike (originalne, sa žigom, i komprimovane sa žigom) sa odgovarajućim referentnim obrascima (levo), za prvu stranu knjige 'Elementa geometriae' (desno).

Donja linija odgovara vrednostima linearne korelacije **originalne slike** sa referentnim obrascima**.**

[31] Ovo važi i za druge slične grafike u radu.

Gornja linija odgovara vrednostima linearne korelacije **slika sa žigom** i njima odgovarajućih referentnih obrazaca.

Linija u sredini odgovara vrednostima linearne korelacije **komprimovanih slika sa žigom** i njima odgovarajućih referentnih obrazaca.

Iz prikazanog grafika vidi se da za svaki referentni obrazac $r(i)$, $(i = 1,...,k)$ važi:

- Ugradnja binarne jedinice povećava vrednost linearne korelacije slike sa referentnim obrascem za α
- Kompresija smanjuje linearnu korelaciju za konstantu, tj. briše konstantni deo žiga
- Posle kompresije, ostaje konstantni deo žiga

(Jasno, pri ugradnji binarne nule, vrednost linearne korelacije se smanjuje za α, a kompresija povećava korelaciju za konstantu – briše konstantni deo žiga).

Za svih 20 referentnih obrazaca rezultat je isti: za α=3, DjVu Photo kompresija je "pojela" oko 2.25 referentnog obrasca. Drugim rečima, efekat ove kompresije je isti kao da je žig ugrađen snagom $\alpha' = 0.75$.

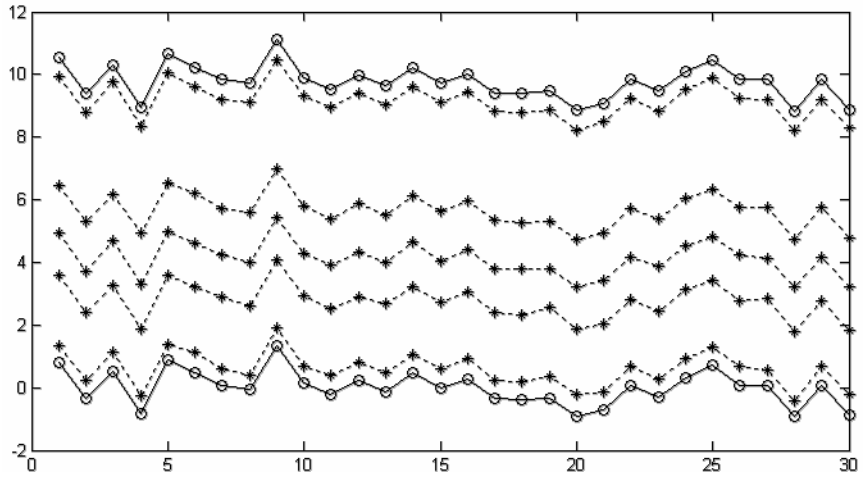
Tako, uvodimo novi pojam: *parametar snage ugradnje posle kompresije*, α'. α' **je** α **koje ostaje posle kompresije.**

Zaključak: Koliki deo žiga će se očuvati, ne zavisi od izbora referentnog obrasca i od linearne korelacije originalne slike sa njim. Ili, **za dato** α, α' **ne zavisi od ključa žiga.**

10.1.2. Zavisnost α' od tehnike kompresije

Isti test ponovljen je i za tehnike kompresije različite od DjVu Photo (kompresija talasićima). Tako, u slučaju JPEG kompresije (koja se zasniva na deobi slike na 8×8 blokove i zatim diskretnoj kosinusnoj transformaciji primenjenoj na svaki blok posebno), zaključak je isti: posle kompresije ostaje isti deo žiga, bez obzira koji ključ je korišćen kao seme generatora referentnog obrasca.

Slika 10.2 pokazuje ovu karakteristiku žiga za JPEG kompresiju. Test je obavljen za sliku 'Cameraman'. Snaga ugradnje je $\alpha = 10$, a JPEG kompresije su 5%, 25%, 45%, 65% i 85%. Linije na slici predstavljaju vrednosti linearne korelacije za slike sa 30 referentnih obrazaca. Donja linija predstavlja ih za originalnu sliku. Gornja je za slike sa žigom. Pet linija između su za komprimovane slike sa žigom (pet snaga kompresije).



Slika 10.1: Vrednosti linearnih korelacija za slike 'Cameraman' (originalnu, sa žigom, i komprimovanu sa žigom) sa odgovarajućim referentnim obrascima, za različite intenzitete JPEG kompresije

Zaključak: I kod JPEG kompresije, za datu sliku i datu snagu kompresije, α' je konstantno za fiksno α (ne zavisi od referentnog obrasca koji je ugrađen).

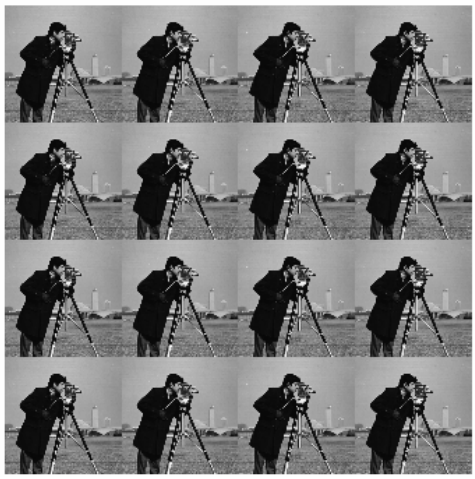
Kod svake od tehnika kompresije za koju je eksperiment vršen, za datu sliku i koeficijent α, ostaje posle kompresije očuvano isto α', nezavisno od ključa žiga i vrednosti linearne korelacije originalne slike sa referentnim obrascem koji se ugrađuje.

Za dato α, vrednost α' je manja kod intenzivnije, nego kod manje intenzivne kompresije. Na primer, α' je veće za JPEG 70% kompresiju nego za JPEG 40%. DjVu Clean kompresija će imati još manju vrednost α'.

10.1.3. Zavisnost α' od veličine slike

U sliku nastalu iz 4×4 jednake slike (slika 10.3) ugrađen je bit poruke snagom α. Slika je zatim izložena kompresiji sa gubicima.

Vrednosti α' za svaku od podslika i za celu sliku su jednake.



Slika 10.3: Slika sastavljena iz 4×4 slike 'Cameraman'

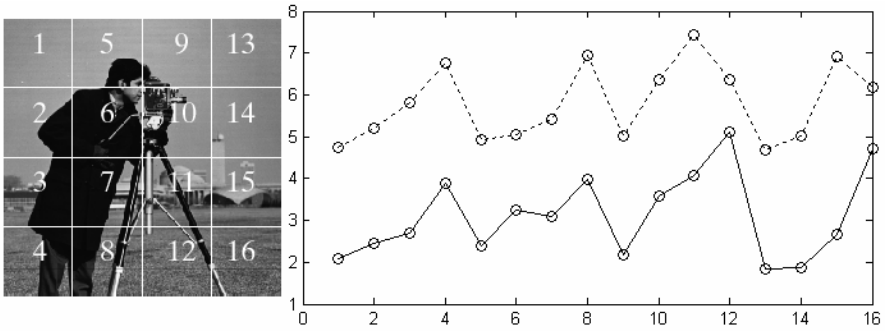
Zaključak: Koeficijent α' ne zavisi od dimenzije slike.

10.1.4. Zavisnost α' od sadržaja slike

U svrhu ispitivanja vrednosti α' (za zadatu vrednost α i tehniku kompresije) za različite slike, obavljen je sledeći test:

U sliku c_0 ('Cameraman') ugrađuje se poruka snagom $\alpha = 7$. **Rezultujuća slika c_w izlaže se zatim (DjVu Photo) kompresiji. Rezultat je komprimovana slika sa žigom c_{wn}.**

Detektor učitava sliku c_{wn} i deli je na 4×4 jednaka dela. Svaki deo (podsliku) tretira kao zasebnu sliku. Za svaku podsliku računaju se vrednosti α' i entropija. Prikazuju se grafikom vrednosti α' i entropije po podslikama (podslike se ređaju kolona po kolona) (slika 10.4)



Slika 10.4: Podslike slike 'Cameraman', i grafici vrednosti entropije (gornja) i α' (donja linija) za 16 delova slike

Vrednost α' zavisi od slike (svaka podslika je slika sa svojim karakteristikama). Vidi se da je (uglavnom) tamo gde je veća entropija, veća i vrednost α'. To znači da će više žiga ostati sačuvano u aktivnim, nego u ravnim oblastima.

Zaključak: α' zavisi od sadržaja slike. **Žig ostaje više očuvan** (α' je veće za isto α) **u aktivnijim slikama** (ili delovima slike).[32]

[32] U nekom algoritmu žiga, u kome se ne bi u sve delove slike smeštala ista količina informacije, ovaj rezultat bi dao dobar putokaz gde najviše podataka treba smestiti: **u aktivne oblasti slike**!

Razlozi su:

- Ljudsko oko slabije vidi promene u aktivnim, nego u ravnim oblastima

- U aktivnim oblastima žig, osim što je neprimetniji, bolje i preživljava kompresiju.

10.1.5. Određivanje snage α za robusnu poruku

Uslovi koji treba da važe za poruku robusnu prema kompresiji dobiće se ako se u formulama (9.6), (9.7') i (9.7") (potpoglavlje 9.1.3 – *Određivanje koeficijenta snage ugradnje* α), svako α zameni sa α' :

- u slučaju fiksne snage ugradnje: $\alpha' = \tau + 3\sigma_{lc}$ (10.1)
- u slučaju podešavanja snage ugradnje:

$$\alpha' = \max(\tau - l, 0) \text{ ako se ugrađuje binarna jedinica,} \quad (10.2')$$

$$\alpha' = \max(\tau + l, 0) \text{ ako se ugrađuje binarna nula} \quad (10.2")$$

Postupak dobijanja α' iz α je "prirodan", jer odgovara redosledu događaja – α je "uzrok", a α' – "posledica":

- ugradi se žig snagom α ;
- komprimuje se slika;
- očita se α' .

Da bi se odredilo suprotno (α iz α'), mora se probati sa različitim vrednostima α . To se čini koristeći žigove načinjene od **proizvoljnog (ali samo jednog) referentnog obrasca**.

Koraci u određivanju α :

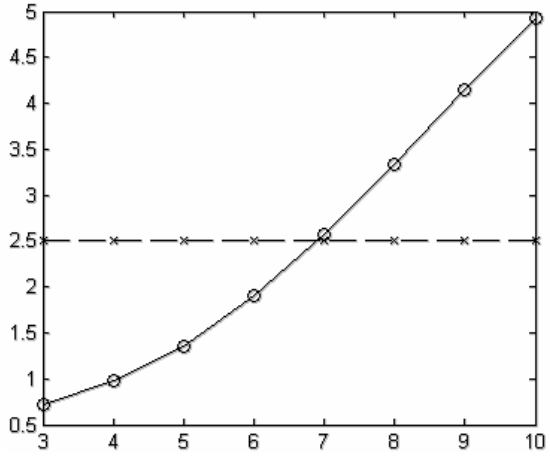
1. Za sliku se odrede τ i α'
2. Ugradi se u sliku žig za nekoliko α vrednosti; na taj način, dobija se nekoliko slika sa žigom.
3. Ove slike sa žigom izlažu se očekivanoj kompresiji.
4. Detektor računa α' za svaku komprimovanu sliku sa žigom. Dobijene vrednosti α' upoređuju se sa željenom vrednošću. Vrednost α čije α' je najbliže željenom je odgovarajuća.

U sledećem testu je jedan referentni obrazac ugrađen u sliku (prva strana knjige 'Elementa geometriae') redom sa α=3, 4, 5, 6, 7, 8, 9 i 10. Slike sa na taj način ugrađenom binarnom jedinicom zatim su izložene DjVu Photo kompresiji. U detekciji je za svaku od ovih slika ispitano koliko poruke je ostalo sačuvano. Cilj ovog testa bio je da se odrede parovi (α, α') za našu sliku.

Rezultati su bili:

α:	3	4	5	6	7	8	9	10
α':	0.72	0.99	1.36	1.90	2.58	3.34	4.14	4.92

Na slici 10.5, α vrednosti za svih osam slika može se pročitati na apscisi. Presek dve linije na ovoj slici pokazuje: da bi α' bilo bar 2.5, treba da je $\alpha \geq 7$.



Slika 10.5: Zavisnost α' od α

10.2. Robusnost dužih poruka prema kompresiji

Ako se u sliku ugrađuje k obrazaca, svaki sa sopstvenim koeficijentom snage ugradnje α_i $(i = 1,...,k)$, biće:

$$c_w \approx c_0 + \sqrt{\alpha_1^{\,2} + \alpha_2^{\,2} + ... + \alpha_k^{\,2}} \cdot r_s = c_0 + \beta r_s, \tag{10.3}$$

$$r_s = \frac{\pm \alpha_1 r_1 \pm \alpha_2 r_2 \pm ... \pm \alpha_k r_k}{\beta} \tag{10.4}$$

Posle kompresije će biti:

$$c_{wn} \approx c_0 + \beta' r_s, \qquad \beta' = \sqrt{(\alpha_1')^2 + (\alpha_2')^2 + ... + (\alpha_k')^2} \tag{10.5}$$

Posle kompresije, koeficijent snage koji preostane biće β' (koeficijent koji odgovara koeficijentu β za datu sliku i tehniku kompresije)

$$p = \beta' / \beta, \qquad \beta' = p \cdot \beta = \sqrt{(p \cdot \alpha_1)^2 + (p \cdot \alpha_2)^2 + ... + (p \cdot \alpha_k)^2} \tag{10.6}$$

Zato,

$$\frac{\alpha_1'}{\alpha_1} = \frac{\alpha_2'}{\alpha_2} = ... = \frac{\alpha_k'}{\alpha_k} = p$$

Procedura za određivanje koeficijenata α_i $(i = 1,2,...,k)$ je:

1. Za sliku se odrede parametri τ i α_i' $(i = 1,2,...,k)$;
2. $\beta' = \sqrt{(\alpha_1')^2 + (\alpha_2')^2 + ... + (\alpha_k')^2}$
3. Nalazi se β iz β' i $p = \frac{\beta'}{\beta}$

4. $\alpha_i = \dfrac{\alpha_i'}{p}$ $(i = 1,2,...,k)$

Primer 10.1: $\alpha_1' = 3, \alpha_2' = 4$

$\beta' = \sqrt{3^2 + 4^2} = 5$

Neka koeficijentu $\beta' = 5$ za datu sliku i tehniku kompresije odgovara koeficijent $\beta = 10$. Tada je

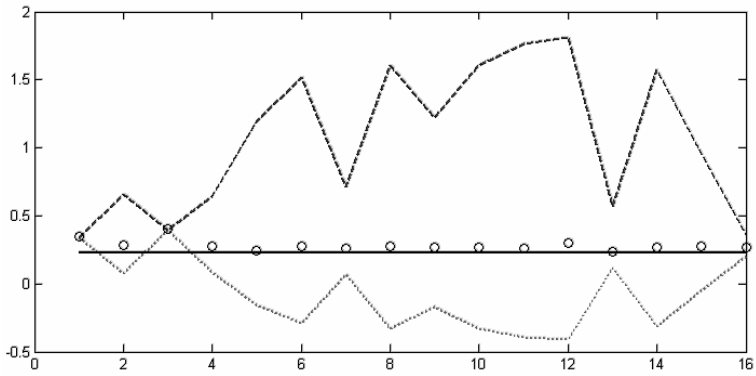
$$p = \frac{\beta'}{\beta} = \frac{1}{2}, \quad \alpha_1 = \frac{\alpha_1'}{p} = 6, \quad \alpha_2 = \frac{\alpha_2'}{p} = 8$$

Primer 10.2: U sliku c_0='Fishingboat' (512×512 piksela), ugrađuje se $k = 16$ bitova poruke (binarnih jedinica).

$$\sigma_{lc} = \frac{\sqrt{E(c_0)}}{(512 \cdot 512)} \approx 0.25 \qquad \tau = 0.25$$

$\alpha' = [0, 0.18, 0, 0.17, 0.41, 0.54, 0.19, 0.58, 0.42, 0.58, 0.64, 0.66, 0.14, 0.57, 0.31, 0.05]$

$\beta' = 1.65, \qquad \beta = 5.4, \qquad p = 0.3 \qquad \alpha_i = \alpha_i' / p \ \ (i = 1,...,k)$



Slika 10.6: Linearna korelacija između slike (originalne, sa žigom i komprimovane sa žigom) i ugrađenih referentnih obrazaca

Slika 10.6 pokazuje vrednosti linearne korelacije između slike i svakog ugrađenog obrasca. Donja linija predstavlja vrednosti linearne korelacije za originalnu sliku; gornja linija je za sliku sa žigom.

Kružići predstavljaju vrednosti linearne korelacije između komprimovane slike sa žigom i ugrađenih referentnih obrazaca. Vidi se da su njihove ordinate bliske vrednosti praga τ. Tako, to su najmanje vrednosti potrebne da bi detektor posle kompresije izvestio da je u sliku ugrađen žig.

Primer 10.3: Isti test je urađen za poruku dugu 64 bita. Žig je takođe preživeo DjVu Photo kompresiju. Ukupna snaga ugradnje je $\beta = 6.94$. Na slici 10.7 mogu se videti komprimovana originalna slika (bez žiga) i komprimovana slika sa žigom.

Slika 10.7: Komprimovan original i komprimovana slika sa žigom ('Fishingboat', 512×512 **piksela,** $k = 64$ **bita poruke, DjVu Photo kompresija)**

11. Ugradnja žiga belog Gausovog šuma u domenu transformacije

U ovom poglavlju opisani algoritam ugradnje će biti primenjen u domenu transformacije (ugradnja i detekcija obavljaju se u domenu transformacije, umesto u prostornom domenu). Biće pokazano da sa gledišta efikasne ugradnje žiga belog Gausovog šuma, nema razlike između ugradnji u prostornom ili domenu transformacije (DCT, blok DCT, Furijeovom, ili domenu neke ortogonalne transformacije talasićima).

Pre svega, većina transformacija koje se koriste u obradi slika i kompresiji, su ortogonalne (ili bar unitarne) linearne transformacije.

Linearna transformacija f očuvava operacije sabiranja i skaliranja. Zato,

$$f(c_0 + \alpha \cdot r) = f(c_0) + \alpha \cdot f(r) \tag{11.1}$$

Ortogonalna transformacija f čuva skalarni proizvod, pa time i dužine vektora i uglove među njima. Zato,

$$\|c_0\| = \|f(c_0)\| \qquad \text{i} \qquad \|r\| = \|f(r)\| \tag{11.2}$$

$$\cos(c_0, r) = \cos(f(c_0), f(r)) \tag{11.3}$$

$$lc(c_0, r) = lc(f(c_0), f(r)) \tag{11.4}$$

Takođe, kod ortogonalnih (i unitarnih) linearnih transformacija važi ***Parsevalova jednakost***:

Ako je $b_0 = f(c_0)$, onda je $E(b_0) = \sum_{i=1}^{mn} (b_0(i))^2 = \sum_{i=1}^{mn} (c_0(i))^2 = E(c_0)$ (11.5)

Ortogonalna transformacija preslikava referentni obrazac $r = (r(1), r(2), ..., r(mn))$ (vektor sa koordinatama iz raspodele $N(0,1)$), u vektor $f(r)$ iz raspodele $N(0,1)$, tj. **ortogonalna transformacija preslikava referentni obrazac u referentni obrazac.**

Standardne devijacije od nule za linearnu korelaciju vektora c_0 i referentne obrasce u prostornom i u domenu transformacije su jednake:

$$\sigma_{lc}(c_0) = \sigma_{lc}(f(c_0)) \tag{11.6}$$

Sasvim je isto da li se u sliku žig belog Gausovog šuma ugrađuje snagom α u prostornom ili u domenu ortogonalne transformacije. Vrednosti korelacije biće iste u oba domena.

Takođe, nema razlike u robusnosti ovakvog žiga prema kompresiji u prostornom i u domenima transformacije.

12. Ugradnja žiga belog Gausovog šuma u podsliku u domenu transformacije

Kao i u prostornom domenu, moguće je ugraditi žig u **deo koeficijenata** u domenu transformacije (ovde se to takođe naziva *ugradnjom u podsliku*).

Tehnički, nema razlike u određivanju snage ugradnje, efikasne i robusne prema kompresiji, između ugradnje u deo koeficijenata (podsliku) i u celu sliku. Snaga efikasne ugradnje određena je dimenzijom i energijom slike (ili dela slike) u koju će žig belog Gausovog šuma biti ugrađen.

Robusnost prema kompresiji određena je svojstvima tih koeficijenata u odnosu na očekivanu kompresiju. Međutim, ispitivanje robusnosti u delu koeficijenata slike u domenu transformacije je znatno složeniji problem od ispitivanja u slučaju ugradnje preko cele slike. Tokom ugradnje često nije ni poznato kojoj kompresiji će slika zatim biti izložena. U takvim okolnostima nije moguće dati generalan savet koji se tiče snage ugradnje.

Mogu se analizirati samo pojedini slučajevi.

Ovde će biti dati neki primeri za takvu ugradnju u blok DCT domenu. Ovaj domen, koji predstavlja osnovu JPEG kompresije, često se koristi u ugradnji digitalnih žigova. Pri opisu takve ugradnje, biće korišćen termin *potkanal slike* (uveli su ga Eggers i Girod u [7_09]). *Potkanal* je vektor koji u blok DCT domenu ima za koordinate – elemente sa istim indeksom u blokovima. Potkanali su uređeni u skladu sa cikcak redosledom. Tako, potkanal 1 se sastoji iz svih DC elemenata blokova slike; potkanal 10 se sastoji od svih elemenata koji su na poziciji 10 u bloku (cikcak redosled).

Slika dimenzije $m \times n$ u 8×8 blok DCT domenu može biti predstavljena sa 64 potkanala: [33]

---

[33] Ovde se pretpostavlja da su dumenzije slike multipli broja 8, da bi podela u 8×8 blokove bila moguća.

$s_j = (s_j(1), s_j(2), ..., s_j(nbl))$, ($j = 1,2,...,64$), gde je $nbl = mn/64$ (12.1)

U sledeća dva potpoglavlja biće ilustrovana sa nekoliko primera, procedura za određivanje koeficijenta snage, za ugradnju u deo koeficijenata u domenu transformacije. Potpoglavlje 12.1 pokazuje kako se određuje koeficijent snage za efikasnu ugradnju u potkanale slike. U potpoglavlju 12.2 ispituje se robusnost prema kompresiji žiga ugrađenog u neke potkanale.

U svrhu ilustracije rezultata koristi se slika 'Cameraman', dimenzije 256x256 piksela.

12.1. Efikasnost ugradnje u potkanale slike

12.1.1. Ugradnja žiga u prva 32 potkanala

Ovde se nalazi mera efikasne ugradnje za prva 32 potkanala

$$\boldsymbol{c}_{ws} = \boldsymbol{c}_0 + \alpha_s \cdot \boldsymbol{r}_s , \tag{12.2}$$

i dobijeni rezultati se porede sa ugradnjom preko cele slike

$$\boldsymbol{c}_w = \boldsymbol{c}_0 + \alpha \cdot \boldsymbol{r} . \tag{12.3}$$

Obrazac $\boldsymbol{r}_s$ dobija se iz referentnog obrasca $\boldsymbol{r}$, anuliranjem podataka u potkanalima 33–64 (jasno, $\boldsymbol{r}_s$ nije više referentni obrazac).

Kod ugradnje preko cele slike c_0, računa se parametar σ_{lc} pomoću formule:

$$\sigma_{lc}(c_0) = \frac{\sqrt{E(c_0)}}{\dim(c_0)} \tag{12.4}$$

Ugradnja u prva 32 potkanala može se posmatrati kao ugradnja u podsliku $\boldsymbol{s} = \{\boldsymbol{s_i}, \boldsymbol{i} = 1,2,...,32\}$, i

$$\sigma_{lc}(s) = \frac{\sqrt{E(s)}}{\dim(s)} \approx \frac{\sqrt{E(c_0)}}{\frac{\dim(c_0)}{2}} = 2 \cdot \sigma_{lc}(c_0) , \tag{12.5}$$

jer je gotovo sva energija slike skoncentrisana u prvih 32 potkanala (na primer, kod slike 'Cameraman', to je čak 99.76% od ukupne energije slike).

Zato, za efikasnu ugradnju, u slučaju ugradnje u prva 32 potkanala, potrebna je dvostruka snaga, ako poredimo sa ugradnjom preko cele slike:

$$\alpha_s = 2 \cdot \alpha \tag{12.6}$$

Da bi se izračunao stvarni uticaj ugradnje žiga na vernost slike, treba primetiti da je $\|\boldsymbol{r}_s\| = \|\boldsymbol{r}\|/\sqrt{2}$. Ako se "standardizuje" obrazac $\boldsymbol{r}_s$ tj. dovede na normu referentnog obrasca $\boldsymbol{r}$, biće

$$\left\|\boldsymbol{r}_s^{\boldsymbol{nor}}\right\| = \sqrt{2}\cdot\|\boldsymbol{r}_s\| \tag{12.7}$$

Zato, efektivna snaga ugradnje je

$$\alpha_s^{\boldsymbol{nor}} = \sqrt{2}\cdot\alpha\,. \tag{12.8}$$

Ugradnja u prva 32 potkanala, da bi bila efikasna, mora biti obavljena $\sqrt{2}$ puta većom snagom, u odnosu na ugradnju preko cele slike.

Takođe, $MSE(c_0, c_{ws}) = 2\cdot MSE(c_0, c_w)$ (12.9)

(da bi ugradnja bila efikasna, potrebno je načiniti dva puta veću srednju kvadratnu grešku u odnosu na originalni algoritam).

Ovakva ugradnja je i uočljivija, jer je poruka ugrađena u nižim frekvencijama.

Sledi ispitivanje kapaciteta i robusnosti za neke potkanale iz prve polovine cikcak skena. Kao i u [7_09], za to se koriste predstavnici. To su, tradicionalno, potkanali 1, 10 i 22.

12.1.2. Ugradnja u potkanal 1

Za sliku 'Cameraman' (dimenzije 256×256), vrednost parametra σ_{lc} za prvi potkanal je:

$$\sigma_{lc\_1} = \frac{\sqrt{E(s_1)}}{nbl} = \frac{\sqrt{1130.41\cdot10^6}}{1024} = 32.83$$

Na osnovu potpoglavlja 9.1.4. (*Određivanje koeficijenta snage ugradnje* α), snaga ugradnje mora biti (za jedan bit poruke, i algoritam ugradnje fiksnom snagom), najmanje $\alpha = \tau + 3\cdot\sigma_{lc\_1}$, tj. najmanja vrednost α, potrebna za efikasnu ugradnju jednog bita poruke, (ako uzmemo $\tau = \sigma_{lc\_1}$) mora biti čak 131.33!

Ako standardizujemo ugrađeni obrazac r_1 (r_1 nastaje kada u referentnom obrascu r anuliramo sve elemente osim u potkanalu 1), biće $\left\|r_1^{nor}\right\| = 8\cdot\|r_1\|$. Zato, standardizovani koeficijent snage ugradnje je $\alpha^{\boldsymbol{nor}} = \alpha/8 = 16.42$. Srednja kvadratna greška za ovu ugradnju je $mse = MSE(c_0, c_w) = 268.92$.

Jasno, ugradnja žiga sa slepom detekcijom u ove koeficijente je izvan diskusije, zbog ekstremno visokih DC koeficijenata.

12.1.3. Ugradnja u potkanal 10

$$\sigma_{lc\_10} = \frac{\sqrt{E(s_{10})}}{nbl} = \frac{\sqrt{1.14 \cdot 10^6}}{1024} = 1.04$$

Za efikasnu ugradnju jednog bita poruke u potkanalu 10 (ako se uzme $\tau = \sigma_{lc\_10}$), treba koristiti snagu ugradnje $\alpha = 4 \cdot \sigma_{lc\_10} = 4.17$, $\alpha^{nor} = \alpha / 8 = 0.52$. Može se ugraditi 64 bita poruke sa $\beta = 8 \cdot \alpha = 33.35$ ($\beta^{nor} = \beta / 8 = 4.17$, $mse = 16.22$). Naizgled, ova vrednost mse i nije prevelika. Međutim, promene u slici su dosta primetne (Slika 12.1). Potkanal 10 sadrži podatke o niskofrekventnoj komponenti slike. HVS je na takve podatke osetljiv. Zato se mora biti oprezan sa snagom ugradnje za ovu komponentu.

Slika 12.1: Originalna i slika posle ugradnje poruke sa $\beta = 33.35$ u potkanalu 10

12.1.4. Ugradnja u potkanal 22

$$\sigma_{lc\_22} = \frac{\sqrt{E(s_{22})}}{nbl} = \frac{\sqrt{0.23 \cdot 10^6}}{1024} = 0.47$$

Za našu sliku, snaga ugradnje (za poruku od jednog bita, fiksnu snagu ugradnje i $\tau = \sigma_{lc\_22}$) je $\alpha = 4 \cdot 0.47 = 1.89$. Zato, snagom ugradnje $\beta = 20$, u taj potkanal se može ugraditi nešto više od 100 bitova, a snagom ugradnje $\beta = 40$ – čak 400. Tako, kapacitet ugradnje u ovom potkanalu nije problem.

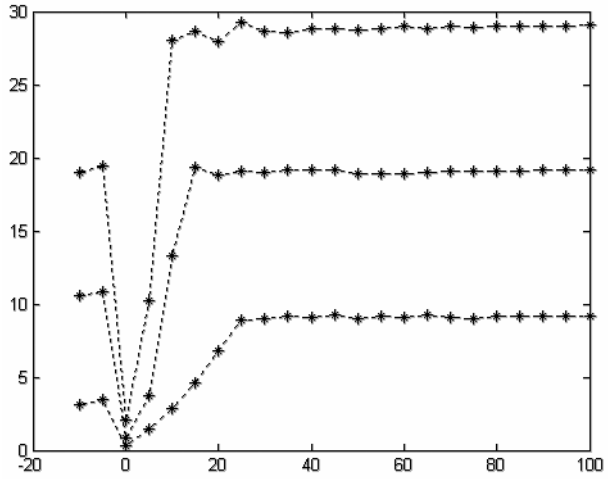
Za $\beta = 20$, $mse = 6.14$; za $\beta = 30$, $mse = 13.80$; za $\beta = 40$, $mse = 24.54$. Čak i poslednja navedena vrednost mse nije prevelika. Ona odgovara snazi ugradnje 5 (kod ugradnje preko cele slike). Ipak, ugradnja u potkanalu 22 je ugradnja u oblasti niske frekvencije, pa je ovde subjektivni kvalitet slike nešto lošiji.

12.2. Robusnost prema kompresiji kod ugradnje u potkanale slike

Sada će za malopre predstavljene primere ugradnje u blok DCT domenu, biti analizirana robusnost žiga prema nekim tehnikama kompresije. Biće ispitan efekat ugradnje u neke koeficijente blok DCT domena – ako slika zatim bude izložena JPEG ili nekoj drugoj kompresionoj tehnici. Među ovim "različitim" tehnikama kompresije, biće predstavljen i efekat DjVu kompresije talasićima.

12.2.1. Potkanal 10

U sledećem eksperimentu, slika sa žigom ugrađenim snagama $\beta = 10$, $\beta = 20$ i $\beta = 30$ izložena je većem broju različitih kompresija: JPEG (od 0% do 100%, sa korakom 5%), i DjVu (Photo i Clean). Rezultati robusnosti žiga za ovaj potkanal prikazani su na slici 12.2. Intenziteti JPEG kompresije su predstavljeni na apscisi. U svrhu predstavljanja i rezultata za DjVu kompresiju na istoj slici, za DjVu Photo koristi se vrednost apscise -5, a za DjVu Clean – vrednost -10.



Slika 12.2: Robusnost potkanala 10 prema različitim kvalitetima kompresije.

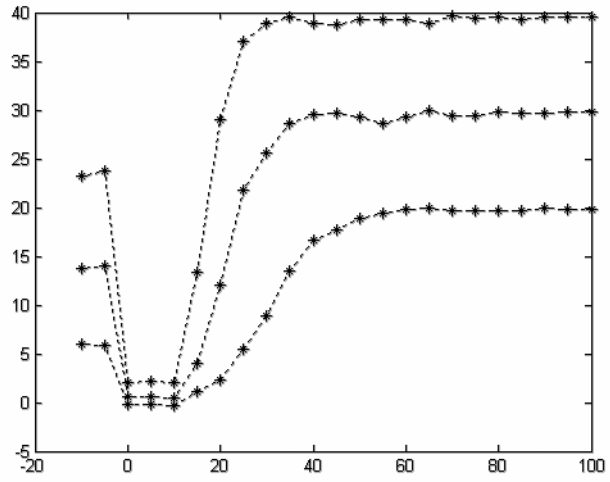
Sa grafika se može videti da je ugradnja u ovaj potkanal robusna prema razumnom intenzitetu JPEG kompresije,[34] i da ugrađeni žig ostaje praktično potpuno neoštećen. Tako, odgovarajuće snažna ugradnja u ove koeficijente je praktično imuna prema svakoj smislenoj JPEG kompresiji. Što se tiče DjVu kompresije, ona uništava deo podataka ugrađenih u ovaj potkanal. Zato treba o tome povesti računa kada se bira snaga

[34] Sa kvalitetom kompresije $< 20\%$, slika je praktično bezvredna. Zato neće biti bitno da li je tada žig očuvan.

ugradnje: da bi se poruka mogla detektovati posle DjVu kompresije, treba je ugraditi odgovarajućom snagom.

12.2.2. Potkanal 22

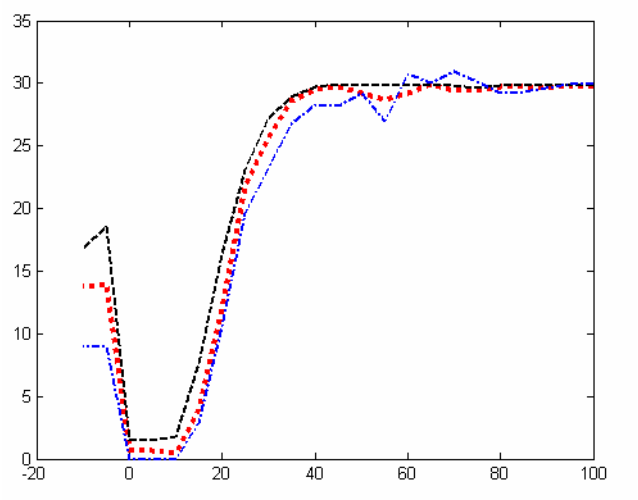
Na slici 12.3 može se videti robusnost potkanala prema snagama ugradnje 20, 30 i 40.
Za slabu snagu ugradnje u potkanalu 22, robusnost prema kompresiji nije dobra. Ako je ukupna snaga ugradnje dovoljno velika, žig će posle kompresije opstati najvećim delom. Na slici, može se videti da za robusnost prema nešto snažnijoj kompresiji, žig treba da bude ugrađen nešto snažnije. Takođe, za DjVu kompresiju, vidi se da, ako se žig ugradi dovoljno snažno, on će posle kompresije ostati detektabilan.



Slika 12.3: Robusnost potkanala 22 prema kompresiji

Važna napomena:

Dobijene vrednosti u ovom poglavlju su manje–više slične za većinu prirodnih slika. Tako, prethodne dve slike su slične za njih. Na Slici 12.4 predstavljene su odgovarajuće vrednosti u potkanalu 22 za snagu ugradnje $\beta = 30$, za slike 'Cameraman' (crveno), prvu stranu knjige 'Elementa geometriae' (crno) i 'ravnu' sliku (svi pikseli imaju istu boju – nijansu sive) (plava linija).



Slika 12.4: Ilustracija jednakog ponašanja poruke belog Gausovog šuma u potkanalu 22, za tri različite slike

13. Žig belog Gausovog šuma i druge modifikacije slike

Uobičajene modifikacije slike klasifikuju se ([4_01]) u dve grupe – valumetrijska i geometrijska izobličenja.

13.1. Valumetrijska izobličenja

Valumetrijska izobličenja su jednostavnija od geometrijskih. Ona menjaju vrednosti pojedinačnih piksela. U njih ubrajamo aditivni šum, promene amplitude, linearno filtriranje i kompresiju sa gubicima.

Aditivni šum

Ovaj napad ima efekat dodavanja slučajnog signala. Za sliku sa žigom, dodavanje šuma je definisano formulom:

$$c_{w1} = c_w + s, \tag{13.1}$$

gde je $c_w = c_0 + \alpha \cdot r$ slika sa žigom, a s je slučajan vektor uzet iz neke raspodele, nezavisan od c_w i referentnog obrasca r.

Jasno,

$$lc(c_{w1}, r) = lc(c_w, r) + lc(s, r) \approx lc(c_w, r) \tag{13.2}$$

(jer je s nekorelirano sa r).

Zato, aditivan šum ne utiče na žig belog Gausovog šuma (žig je robustan u odnosu na ovaj napad).

Ovaj žig je takođe robustan i prema **promeni sjajnosti** ($c_{w1} = c_w + n \cdot J$, gde je J matrica jedinica, a n ceo broj).

Promena amplitude

Može biti predstavljena formulom

$$c_{w1} = \nu \cdot c_w, \tag{13.3}$$

gde je $\nu > 0$ faktor skaliranja. Ovakva operacija uzrokuje **promenu sjajnosti i kontrasta slike.**

Vrednost linearne korelacije biće jednaka polaznoj, pomnoženoj faktorom ν. U zavisnosti od faktora ν, vrednost linearne korelacije (detektabilnost žiga) će se povećati (ako $\nu > 1$), ili smanjiti (ako $\nu < 1$).

Linearno filtritanje

Može biti dato formulom

$$c_{w1} = c_w * f, \tag{13.4}$$

gde je f filter, a $*$ označava konvoluciju. Mnoge uobičajene modifikacije slika obavljaju se koristeći linearne filtre. Primeri su efekti zamućenja i izoštravanja (**blurring, sharpening)**.

13.2. Geometrijska izobličenja

Ova klasa napada uključuje mnoga izobličenja slika, kao što su **rotacija, prostorno skaliranje, translacija, iskošavanje, opsecanje, transformacija perspektive i promene dimenzija slike.**

Ovi napadi su znatno komplikovaniji od valumetrijskih, zato što dislociraju informaciju o pikselima u matrici slike. Takođe, oni obično menjaju dimenzije matrice slike. Zato ovde žig nije moguće odmah detektovati. Ipak, ovim napadima informacija o ugrađenoj poruci neće biti izgubljena, nego samo "**maskirana**".

Kod svakog geometrijskog napada, pre detekcije, potrebno je obaviti odgovarajuću **proceduru restauracije**. Ova procedura nije jedinstveno određena, i ona je van teme ovog teksta.

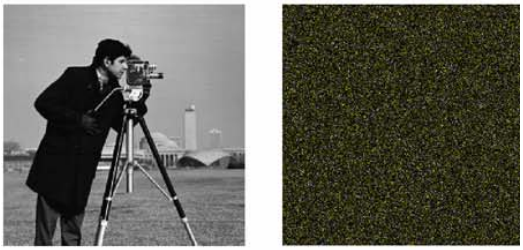
U svrhu ilustracije, ovde će biti opisan jedan geometrijski napad – rotacija slike za ugao φ. Većina zaključaka može se primeniti na druge napade.

13.2.1. Robusnost prema rotaciji

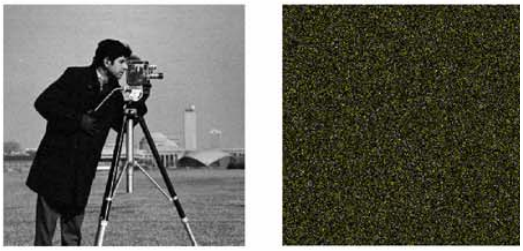
Rešavanju problema u detekciji, nastalih zbog rotacije slike, moguće je pristupiti na različite načine. Tako, ako je slika sa žigom rotirana za ugao φ, da bi detekcija postala moguća, može se uraditi nešto od sledećeg:

- (Rotirana) slika može se porediti (koristeći linearnu korelaciju) sa referentnim obrascem koji je takođe rotiran za ugao φ

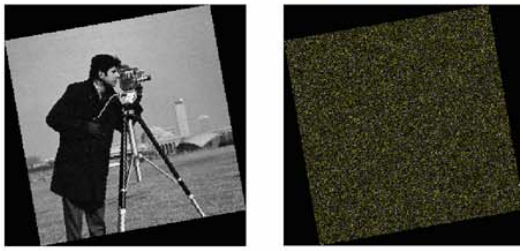
- Pre detekcije, može se rotirana slika rotirati za ugao $-\varphi$, pa porediti sa referentnim obrascem, koji je takođe rotiran za uglove φ i $-\varphi$
- Slika rotirana za ugao φ pa za $-\varphi$ (i zatim opsečena na dimenzije originalne slike), može se porediti sa originalnim referentnim obrascem.



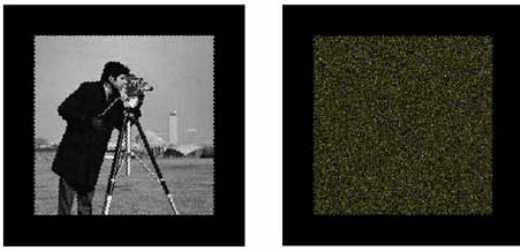
c_0 (256,256) – originalna slika ('Cameraman')
r (256,256) – referentni obrazac
$lc_0 = lc(c_0, r) = 0.787$



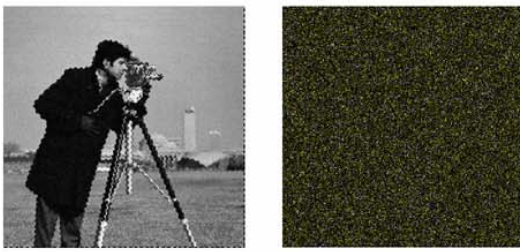
c_w (256,256) – slika sa žigom (snaga ugradnje $\alpha = 5$)
r (256,256) – referentni obrazac
$lc_w = lc(c_w, r) = 5.787$



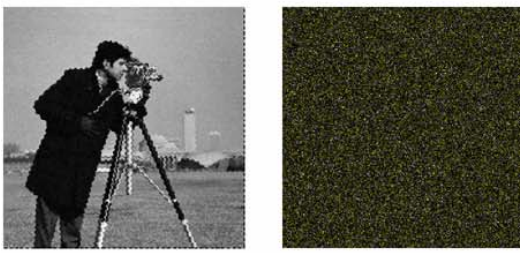
c_{wr} (297,297) – slika sa žigom, rotirana za ugao $\varphi(=10^o)$
r_r (297,297) – referentni obrazac, rotiran za φ
$lc_r = lc(c_{wr}, r_r) = 5.815$



c_{wr1} (345,345) – slika sa žigim, rotirana za φ, pa zatim $-\varphi$
r_{r1} (345,345) – referentni obrazac, rotiran za φ, pa $-\varphi$
$lc_{r1} = lc(c_{wr1}, r_{r1}) = 5.591$



c_{wr2} (256,256) – slika sa žigom, rotirana za φ, pa $-\varphi$, posle opsecanja na dimenzije originalne slike
r_{r2} (256,256) – referentni obrazac, rotiran za φ i $-\varphi$, posle opsecanja na originalne dimenzije
$lc_{r2} = lc(c_{wr2}, r_{r2}) = 5.592$



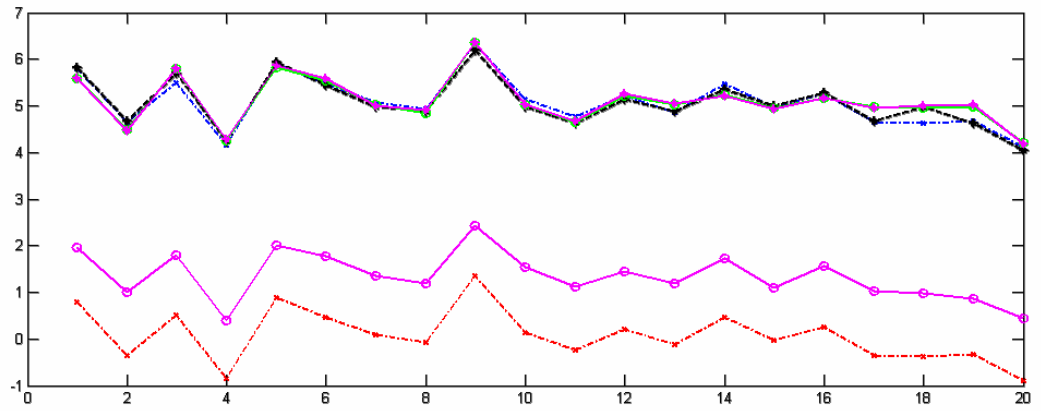
c_{wr2} (256,256) – slika sa žigom, rotirana za φ, pa $-\varphi$, opsečena na dimenzije originalne slike
r (256,256) – referentni obrazac
$lc_{r2p} = lc(c_{wr2}, r) = 1.953$

Slika 13.1: Vrednosti linearne korelacije za sliku i referentni obrazac (rotirane i nerotirane)

Na slici 13.1 prikazane su vrednosti linearne korelacije za sliku i referentni obrazac (nerotirane i rotirane). U svakom redu su predstavljeni slika i obrazac za koje se računa korelacija. Tako, za svaki primer je navedeno koja slika i obrazac (i kojih dimenzija) su u pitanju, i takođe vrednost linearne korelacije za njih.

Rotacija praktično ne smanjuje vrednost linearne korelacije, ako se računa za na isti način rotirane sliku i referentni obrazac. Međutim, ako se poredi rotirana slika (posle restauracije na origunalnu poziciju, rotacijom u suprotnom smeru i opsecanjem na originalne dimenzije) sa originalnim referentnim obrascem, izvesna količina žiga biće izgubljena (vrednost α' biće manja od vrednosti snage ugradnje α). Treba imati u vidu da je u slučajevima kada nisu poznati svi detalji nastanka izobličenja na slici, ovo drugo rešenje je obično jedino moguće.

Na slici 13.2 predstavljene su ovde spomenute vrednosti linearne korelacije za 20 različitih referentnih obrazaca, za sliku 'Cameraman', $\varphi = 10^o$ i $\alpha = 5$. Može se videti da (slično kao u ranije analiziranom napadu kompresije), za datu sliku i snagu ugradnje α, posle rotacije za dati ugao φ, ostaje snaga α', koja **ne zavisi od referentnog obrasca, niti od njegove korelacije sa originalnom slikom**. Tako, za odluku kojom snagom α treba ugraditi žig u sliku, da bi posle rotacije ostala snaga α', **dovoljno je eksperimentisati samo sa jednim referentnim obrascem.**



Slika 13.2: Vrednosti lc_0 (donja linija), $lc_w, lc_r, lc_{r1}, lc_{r2}$ (gornje linije) i lc_{r2p} (linija u sredini) za 20 referentnih obrazaca

Navedeno razmatranje važi i za druga (valumetrijska i geometrijska) izobličenja. Procedura određivanja neophodne snage ugradnje α obavlja se u sledećim koracima:

(1) Kao i u slučaju ugradnje žiga koji treba da bude robustan prema kompresiji sa gubicima, primenjuju se formule (10.1), odnosno (10.2') i (10.2''), da bi se odredila snaga neophodna za detektabilnost žiga neposredno po ugradnji (α').

(2) Eksperimentišući **samo** sa **jednim referentnim obrascem**, treba odrediti neophodnu snagu ugradnje α, takvu da posle očekivanog napada, snaga koja preostane bude bar α'.

Zaključci i budući rad

U ovom radu razmatran je skup algoritama ugradnje žiga belog Gausovog šuma u sliku (u nijansama sive boje). Tražena je potrebna snaga, koja osigurava detektabilnost žiga.

- Data je formula za optimalnu snagu ugradnje (minimalne koja garantuje detektabilnost) za efikasnu ugradnju,
- Dat je algoritam za određivanje optimalne snage ukoliko se očekuje da će od ugradnje do detekcije slika biti izložena nekoj (očekivanoj) modifikaciji: pre svega kompresiji, ali možda i nekom drugom izobličenju (rotaciji, opsecanju, promeni sjajnosti i kontrasta,...)
- Analizirani su slučajevi ugradnje žiga (belog Gausovog šuma) u celu sliku i u podslike, u prostornom i domenu transformacije i robusnost takve ugradnje prema raznim slučajevima kompresije.
- Kod drugih napada, procedure za određivanje snage α su slične kao u slučaju kompresije. Kod geometrijskih napada, međutim, potrebno je prvo obaviti određene intervencije (restauraciju) da bi takva detekcija bila moguća.

U ovom trenutku, u toku su moja istraživanja u vezi sa žigom belog Gausovog šuma i Krejverovim (Craver) napadom (videti više u potpoglavlju 6.5 – *Sigurnost žiga*).
Moji dosadašnji rezultati ukazuju na činjenicu da **žigovi belog Gausovog šuma nisu invertibilni**, i da se zato mogu koristiti u dokazu vlasništva u sudskom sporu. Mada je u ovom momentu prerano o tome iznositi detalje, može se reći da je izvesnom **modifikacijom detektora, zasnovanom na kompresiji sa gubicima**, moguće dokazati da li je u sliku određeni žig zaista ugrađen ili nije.
Druga interesantna tema je informisano kodiranje, zasnovano na **Kostinoj tehnici prljavog papira** (potpoglavlje 7.1 – *Informisana i slepa detekcija i ugradnja*). U toku je moje traganje za optimalnom snagom ugradnje žiga belog Gausovog šuma, ako se za svaki bit poruke koristi više različitih kodova.

Dodatak: Matematičke osnove

Tema ove disertacije nije "čisto računarska". Ugradnja belog Gausovog šuma u sliku i kompresija sa gubicima iziskuju poznavanje određenih matematičkih oblasti, pre svega linearne algebre i teorije verovatnoće.

U ovom dodatku dat je kratak prikaz matematičkih znanja potrebnih za razumevanje teksta i ukazano je na kontekst u kome su ta matematička znanja u ovom radu korišćena.

D.1 Skalarni proizvod i norma vektora

Vektorski prostor R^k ($k \in N$) je skup svih uređenih k- torki $x = (x_1, x_2, ..., x_k)$ ($x_i \in R$, $i = 1,2,...,k$, R je polje realnih brojeva)

Uvođenjem ***skalarnog proizvoda*** – preslikavanja

$(x, y) \to x \bullet y$

skupa $R^k \times R^k$ u polje R, tako da za proizvoljne vektore $x, y, z \in R^k$ i svako $a \in R$ važi:

(S1) $(x + y) \bullet z = x \bullet z + y \bullet z$

(S2) $(\alpha \cdot x) \bullet y = \alpha \cdot (x \bullet y)$

(S3) $x \bullet y = y \bullet x$

(S4) $x \neq o \quad x \bullet x > 0$

dobijamo ***euklidski*** vektorski prostor $(R^k, \bullet)$

Skalarni proizvod definisan sa

$$x \bullet y = x_1 \cdot y_1 + x_2 \cdot y_2 + ... + x_k \cdot y_k \tag{D.1}$$

nazivamo ***standardnim skalarnim proizvodom***.

(Euklidska) norma (dužina, intenzitet) vektora x data je sa

$$\|x\| = \sqrt{x \bullet x} = \sqrt{x_1 \cdot x_1 + x_2 \cdot x_2 + ... + x_k \cdot x_k} = \sqrt{x_1^{\,2} + x_2^{\,2} + ... + x_k^{\,2}} \tag{D.2}$$

U vektorskom prostoru C^k (C je polje kompleksnih brojeva) uvodimo (***hermitski) skalarni proizvod*** kao preslikavanje $(x, y) \to x \bullet y$ skupa skupa $C^k \times C^k$ u polje C, tako da su zadovoljene aksiome (S1), (S2), (S4) i

(S3') $x \bullet y = \overline{y \bullet x}$

Skalarni proizvod definisan sa

$$x \bullet y = x_1 \cdot \overline{y_1} + x_2 \cdot \overline{y_2} + ... + x_k \cdot \overline{y_k}\,, \tag{D.3}$$

nazivamo ***standardnim (hermitskim) skalarnim proizvodom***.

***

Slika u nijansama sive, veličine $m \times n$ piksela, u računaru se predstavlja kao realna matrica dimenzije $m \times n$, ili kao vektor dimenzije $m \cdot n$. Svi elementi matrice (vektora) slike su celi brojevi iz skupa $\{0,1,...,255\}$. Tako, skup svih mogućih matrica slika dimenzije $m \times n$ predstavlja podskup (ne i potprostor!) vektorskog prostora R^{mn}. Slično, skup referentnih obrazaca (vektori iz standardne normalne raspodele) iste dimenzije predstavlja podskup vektora prostora R^{mn}. Zato se i na vektore iz ovih skupova mogu primeniti malopre navedene definicije skalarnog proizvoda i norme.

U radu sa slikama koristi se i pojam ***energije slike***. Energija se definiše kao kvadrat euklidske norme:

$$E(x) = \|x\|^2 = x_1^{\,2} + x_2^{\,2} + ... + x_k^{\,2} \tag{D.4}$$

Ustvari, pojam energije je u ovom radu u upotrebi i više od pojma norme, jer se koristi ne samo za celu sliku, nego i za njenu podsliku (podslika je bilo koji podskup piksela slike): energija podslike je suma kvadrata svih njenih elemenata.

D.2 Linearno preslikavanje

Linearno preslikavanje (***linearna transformacija***, ***linearni operator***) je funkcija između dva vektorska prostora, koja **očuvava operacije sabiranja vektora i skalarnog množenja**.

Definicija linearnog preslikavanja:

Neka su V i W vektorski prostori nad istim poljem K. Funkcija $f: V \to W$ je linearno preslikavanje ako za svaka dva vektora $x, y \in V$ i svaki skalar $a \in K$ važe sledeća dva uslova:

$f(x+y) = f(x) + f(y)$ aditivnost

$f(a \cdot x) = a \cdot f(x)$ homogenost

***

U ovoj disertaciji se o linearnom preslikavanju pre svega govori u poglavlju 3 (*Kompresija sa gubicima*) i u poglavlju 11 (*Ugradnja žiga belog Gausovog šuma u domenu transformacije*) Naime, obavezan korak u kompresiji slike (sa gubicima) je linearna transformacija.

Na primer, kod JPEG kompresije se na svaki blok slike primenjuje diskretna kosinusna transformacija (DCT), data formulama (3.1) i (3.2) (direktna i obrnuta DCT). DCT prevodi blok iz prostornog domena (vektor iz vektorskog prostora R^{64}) u blok u domenu transformacije (takođe vektor iz prostora R^{64}). Tako, ovde se radi o linearnom preslikavanju $f: R^{64} \to R^{64}$ (ili preciznije, $\{0,1,...,255\}^{64} \to R^{64}$).

Diskretna Furijeova transformacija (DFT) (formule (3.6) i (3.7)) linearno preslikava matrice – vektore slika (dimenzije $m \times n$) u kompleksne matrice – vektore iste dimenzije (C^{mn}).

D.3 Ortogonalna matrica. Ortogonalna transformacija

Realna kvadratna ***matrica*** M je ***ortogonalna*** ako važi

$M \cdot M^T = M^T \cdot M = I$, tj. $M^{-1} = M^T$ (inverzna matrica jednaka je transponovanoj).

Svaka ortogonalna matrica M definiše ***ortogonalnu transformaciju***. Za svaki vektor $x = (x_1, x_2, ..., x_m)^T \in R^k$ važi:

$y = M \cdot x$ (direktna transformacija) i

$x = M^T \cdot y = M^{-1} \cdot y$ (inverzna transormacija)

($y = (y_1, y_2, ..., y_m)^T \in R^k$)

Ako je prostor kompleksan (C^k umesto R^k), matrica je ***unitarna*** (važi $M^{-1} = \overline{M}^T$) i određuje ***unitarnu transformaciju***.

Unitarna (pa i ortogonalna) transformacija je **linearna i čuva skalarni proizvod**, tj. za svaka dva vektora x', x'' važi

$$Mx' \bullet Mx'' = x' \bullet x'' \tag{D.5}$$

***

U disertaciji, DCT je primer ortogonalne, a DFT – unitarne transformacije. Ustvari, pošto se transformacija slike primenjuje na matricu slike (a ne na jednodimenzioni vektor), ona je kombinacija dve transformacije – po vrstama, pa po kolonama (ili – svejedno je! – obrnuto).

Tako za DCT važi (formula (3.1)):

$$D(u,v) = \frac{2}{\sqrt{mn}} C(u)C(v) \sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y) \cos\frac{\pi(u-1)(2x-1)}{2m} \cos\frac{\pi(v-1)(2y-1)}{2n}$$

($u = 1,...,m$, $v = 1,...,n$), ($C(u) = 1/\sqrt{2}$ ako $u = 1$, inače $C(u) = 1$)

Ovu formulu možemo jednostavnije zapisati (uvodeći odgovarajuće nove oznake) kao

$$D(u,v) = \sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y) \cdot M(u,x) \cdot N(v,y) = \sum_{x=1}^{m} M(u,x) \cdot \sum_{y=1}^{n} S(x,y) \cdot N(v,y) \tag{D.6}$$

ili, u matričnom obliku,

$$D = M^{-1} \cdot S \cdot N \tag{D.7}$$

Matrice M i N su ortogonalne, pa je zato

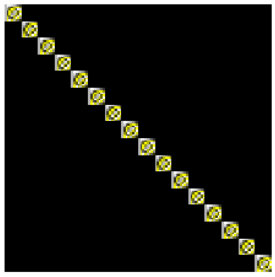
$$D = M^T \cdot S \cdot N \tag{D.8}$$

Jasno, zbog asocijativnosti množenja, važi $D = M^T \cdot (S \cdot N) = (M^T \cdot S) \cdot N$ – matrica D se dobija iz S primenom (kompozicijom) dve ortogonalne transformacije (po vrstama i po kolonama).

Slično razmatranje važi za slučaj (unitarne!) Furijeove transformacije.

Kod JPEG kompresije, navedena formula (3.1) primenjuje se na svaki poseban blok veličine 8×8. Tako, DCT primenjena na svaki blok je kompozicija dve ortogonalne transformacije.

Takođe, ako ovu transformaciju po blokovima gledamo kao jednu **transformaciju, primenjenu na celu sliku**, vidimo da je i ona **ortogonalna**. Ako je slika S dimenzije $m \times n$ (m i n su multipli broja 8), matrica slike će se s leva množiti matricom $M_C{}^T$ (dimenzije $m \times m$), a s desna sa N_C (dimenzije $n \times n$) (ove dve matrice su takođe ortogonalne). Matrica M_C (odnosno N_C) ima po dijagonali malopre navedene ortogonalne matrice M dimenzije 8×8 - svi ostali elementi su nule (videti sliku D.1).



Slika D.1 Matrica M_C dimenzije 128×128

D.4 Normalna raspodela

Definicija: Za slučajnu promenljivu X kažemo da ***ima normalnu raspodelu*** $N(\mu, \sigma^2)$ ako je njen zakon verovatnoća dat funkcijom

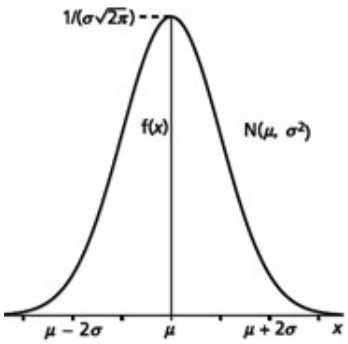
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)} = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right) \tag{D.9}$$

Očekivana vrednost normalne raspodele je μ, a ***standardna devijacija*** σ.

Ova funkcija je simetrična u odnosu na pravu $x = \mu$. U tački $x = \mu$ ima maksimum:

$$f(\mu) = \frac{1}{\sigma\sqrt{2\pi}} \tag{D.10}$$

Grafički prikaz funkcije dat je na slici D.2.



Slika D.2: Normalna raspodela

Specijalno za $\mu = 0$ i $\sigma = 1$, dobićemo ***standardnu normalnu raspodelu***.

***

U disertaciji se govori o digitalnom vodenom žigu u obliku belog Gausovog šuma (slici se dodaje beli Gausov šum – vektor sa elementima koji podležu normalnoj (Gausovoj) raspodeli sredine 0.

Rečnik korišćenih stručnih termina

Srpsko-engleski

(aditivni) beli Gausov šum	additive white Gaussian noise, AWGN
aritmetičko kodiranje	arithmetic coding
bit najmanje težine	least significant bit, LSB
cikcak redosled	zig-zag order
detekcija (digitalnog vodenog žiga)	(digital watermark) detection
detektor (digitalnog vodenog žiga)	(digital watermark) detector
digitalizacija	digitization
(digitalni vodeni) žig	digital watermark
(digitalni vodeni) žig belog Gausovog šuma	AWGN (digital) watermark
(digitalni) rad	(digital) work, multimedia object
diskretna Furijeova transformacija	discrete Fourier transform, DFT
diskretna kosinusna transformacija	discrete cosine transform, DCT
diskretna transformacija talasićima	discrete wavelet transform, DWT
diterovanje, podrhtavanje	dithering
domen transformacije	transform domain
efikasna ugradnja	effective embedding
geometrijsko izobličenje	geometric distortion
greška lažnog negativnog	false negative error
greška lažnog pozitivnog	false positive error
Hafmenovo kodiranje	Huffman coding
informisana ugradnja – detekcija	informed embedding – detection
informisano kodiranje	informed coding
izoštravanje (slike)	(image) sharpening
jedva primetna razlika	just noticeable difference, JND
kôd prljavog papira	dirty paper code
kodiranje dužinom sekvence	run length encoding, RLE
kompresija (sa gubicima)	lossy compression
kompresija bez gubitaka	lossless compression
kompresioni artifakt / artefakt	compression artifact
ljudski vizuelni sistem	human visual system, HVS

lomljiv (digitalni vodeni) žig	fragile (digital) watermark
matični talasić	mother wavelet
niskopropusni filter	low-pass filter
normalna (=Gausova) raspodela	normal (Gaussian) distribution
odnos signal-šum	signal-to-noise ratio, SNR
odnos širine i visine (slike)	(image) aspect ratio
piksel	pixel (picture element)
potkanal slike	image sub-channel
pozadina	background
prag detekcije	detection threshold
praksa (digitalnog vodenog) žiga	digital watermarking
prednji plan	foreground
progresivni prikaz	progressive display
prostorni domen	spatial domain
rečničko kodiranje	dictionary coding
robusna ugradnja	robust embedding
robusnost žiga	watermark robustness
robustan (digitalni vodeni) žig	robust (digital) watermark
sjajnost	brightness
slepa ugradnja – detekcija	blind embedding – detection, oblivious embedding – detection
slika u nijansama sive (boje)	grayscale image
snaga ugradnje	embedding strength
spektar snage	power spectrum
srednja kvadratna greška	mean square error, MSE
stono izdavaštvo	desktop publishing
talasić	wavelet
tehnika raširenog spektra	spread spectrum technique
ugrađivač (digitalnog vodenog žiga)	(digital watermark) embedder
ugradnja (digitalnog vodenog žiga)	(digital watermark) embedding
uzorkovanje	sampling
valumetrijsko izobličenje	valumetric distortion
visokopropusni filter	high-pass filter
zamućenje (slike)	(image) blurring

Englesko-srpski

additive white Gaussian noise, AWGN	(aditivni) beli Gausov šum
arithmetic coding	aritmetičko kodiranje
AWGN (digital) watermark	(digitalni vodeni) žig belog Gausovog šuma
background	pozadina
blind embedding – detection, oblivious embedding – detection	slepa ugradnja – detekcija
brightness	sjajnost
compression artifact	kompresioni artifakt / artefakt
desktop publishing	stono izdavaštvo
detection threshold	prag detekcije
dictionary coding	rečničko kodiranje
digital watermark	(digitalni vodeni) žig
digital watermarking	praksa (digitalnog vodenog) žiga
(digital watermark) detection	detekcija (digitalnog vodenog žiga)
(digital watermark) detector	detektor (digitalnog vodenog žiga)
(digital watermark) embedder	ugrađivač (digitalnog vodenog žiga)
(digital watermark) embedding	ugradnja (digitalnog vodenog žiga)
(digital) work, multimedia object	(digitalni) rad
digitization	digitalizacija
dirty paper code	kôd prljavog papira
discrete cosine transform, DCT	diskretna kosinusna transformacija
discrete Fourier transform, DFT	diskretna Furijeova transformacija
discrete wavelet transform, DWT	diskretna transformacija talasićima
dithering	diterovanje, podrhtavanje
effective embedding	efikasna ugradnja
embedding strength	snaga ugradnje
false negative error	greška lažnog negativnog
false positive error	greška lažnog pozitivnog
foreground	prednji plan
fragile (digital) watermark	lomljiv (digitalni vodeni) žig
geometric distortion	geometrijsko izobličenje
grayscale image	slika u nijansama sive (boje)
high-pass filter	visokopropusni filter

Huffman coding	Hafmenovo kodiranje
human visual system, HVS	ljudski vizuelni sistem
image sub-channel	potkanal slike
(image) aspect ratio	odnos širine i visine (slike)
(image) blurring	zamućenje (slike)
(image) sharpening	izoštravanje (slike)
informed coding	informisano kodiranje
informed embedding – detection	informisana ugradnja – detekcija
just noticeable difference, JND	jedva primetna razlika
least significant bit, LSB	bit najmanje težine
lossless compression	kompresija bez gubitaka
lossy compression	kompresija (sa gubicima)
low-pass filter	niskopropusni filter
mean square error, MSE	srednja kvadratna greška
mother wavelet	matični talasić
normal (Gaussian) distribution	normalna (=Gausova) raspodela
pixel (picture element)	piksel
power spectrum	spektar snage
progressive display	progresivni prikaz
robust (digital) watermark	robustan (digitalni vodeni) žig
robust embedding	robusna ugradnja
run length encoding, RLE	kodiranje dužinom sekvence
sampling	uzorkovanje
signal-to-noise ratio, SNR	odnos signal-šum
spatial domain	prostorni domen
spread spectrum technique	tehnika raširenog spektra
transform domain	domen transformacije
valumetric distortion	valumetrijsko izobličenje
watermark robustness	robusnost žiga
wavelet	talasić
zig-zag order	cikcak redosled

Literatura

Poglavlje 1: **Digitalizacija slika**

[1_01] I.T. Young, J.J. Gerbrands, L.J. van Vliet: *Image Processing Fundamentals*, http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip.html (20.08.2009)

[1_02] John C. Russ: *The Image Processing Handbook,* CRC Press, 2002

[1_03] Jae S.Lim: Two–dimensional signal and image processing, Prentice Hall Ptr, 1990

[1_04] William K. Pratt: *Digital Image Processing: PIKS Inside*, John Wiley &Sons, Inc, 2001

[1_05] Alvy Ray Smith: *A Pixel Is Not A Little Square...,*Microsoft echnical Memo 6, July 17, 1995, http://www.alvyray.com/memos/6_pixel.pdf (4.10.2009)

[1_06] Vesna Vučković: *Digitalizacija slikovnih podataka*, Pregled Nacionalnog centra za digitalizaciju, 2 (2003), 8–16, http://www.ncd.matf.bg.ac.yu/casopis/02/d003/download.pdf (20.08.2009)

[1_07] Vesna Vučković*: Image and its matrix, matrix and its image*, Review of the National Center for Digitization, Issue: 12/2008, http://www.ncd.matf.bg.ac.yu/casopis/12/NCD12017.pdf (20.08.2009)

Poglavlje 2: **Kompresija podataka**

[2_01] D.A. Lelewer, D.S. Hirschberg, *Data compression,* Computing Surveys **19**,3 (1987) 261–297. Reprinted in Japanese BIT Special issue in Computer Science (1989) 165–195, http://www.ics.uci.edu/~dan/pubs/DataCompression.ps.gz (26.09.2009)

[2_02] Guy E. Blelloch: *Introduction to Data Compression*, 2001, http://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/compression.pdf (26.09.2009)

[2_03] C.E.Shannon: *A Mathematical Theory of Communication*, (Reprinted from) The Bell System Technical Journal, Vol. 27, pp 379–423, 623–656, 1948, http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf (21.08.2009)

[2_04] Thomas D. Schneider: *Information Theory Primer*, 2003,

http://people.seas.harvard.edu/~jones/cscie129/papers/nih_info_paper.pdf (20.08.2009)

[2_05] D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098–1102, http://compression.ru/download/articles/huff/huffman_1952_minimum-redundancy-codes.pdf (26.09.2009)

[2_06] A. Said, *Introduction to Arithmetic Coding Theory and Practice,* Hewlett–Packard Laboratories Report, HPL–2004–76, April 2004, http://www.hpl.hp.com/techreports/2004/HPL-2004-76.pdf

[2_07] Howard, Paul G. and Vitter, Jeffrey Scott: *Arithmetic Coding for Data Compression (1992)* https://eprints.kfupm.edu.sa/25621/1/25621.pdf (26.09.2009)

[2_08] Ian H. Witten, Radford M. Neal, John G. Cleary: *Arithmetic coding for data compression*, Communications of the ACM 1987 Volume 30 Number 6, 520–540, http://www.stanford.edu/class/ee398a/handouts/papers/WittenACM87ArithmCoding.pdf (26.09.2009)

[2_09] *RLE compression*, http://www.prepressure.com/techno/compressionrle.htm (21.08.2009)

[2_10] Jacob Ziv, Abraham Lempel: *Compression of Individual Sequences via Variable–Rate Coding*, IEEE Transactions on Information Theory, vol it–24, no. 5, september 1978, 530–536, http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1978_variable-rate.pdf (26.09.2009)

[2_11] T.A. Welch, *A Technique for High–Performance Data Compression*, Computer, vol. 17, no. 6, pp. 8–19, June 1984, doi:10.1109/MC.1984.1659158, http://www.cs.duke.edu/courses/spring03/cps296.5/papers/welch_1984_technique_for.pdf (26.09.2009)

[2_12] Mark Nelson: *LZW Data Compression,* Dr. Dobb's Journal October, 1989, http://www.dogma.net/markn/articles/lzw/lzw.htm (20.08.2009)

Poglavlje 3: **Kompresija sa gubicima**

[3_01]	Yun Q.Shi, Huifang Sun: Image and Video Compression for Multimedia Engineering, CRC Press, 2000
[3_02]	DebugMode Image Compression Toolbox v1.2, http://www.debugmode.com/imagecmp/icomptbx.htm (20.08.2009)
[3_03]	G. K. Wallace, G. K., *The JPEG Still Picture Compression Standard,* IEEE Transactions on Consumer Electronics, December 1991, http://white.stanford.edu/~brian/psy221/reader/Wallace.JPEG.pdf (4.10.2009)
[3_04]	Mark D. Schroeder: JPEG Compression Algorithm and Associated Data Structures, 1997, http://akbar.marlboro.edu/~mahoney/courses/Fall01/computation/compression/jpeg/jpeg.html (4.10.2009)
[3_05]	Hugo Hedberg, Peter Nilsson: A Survey of Various Discrete Transforms used in Digital Image Compression Algorithms,

[3_06]	Andrew B. Watson: *Image Compression Using the Discrete Cosine Transform*, Mathematica Journal, 4(1), 1994, p. 81–88, http://vision.arc.nasa.gov/publications/mathjournal94.pdf (21.08.2009)
[3_07]	Smith, Steven W. (1999): *The Scientist and Engineer's Guide to Digital Signal Processing* (Second ed.) (*Chapter 8: The Discrete Fourier Transform*). California Technical Publishing, http://www.dspguide.com/CH8.PDF (28.09.2009)
[3_08]	Weisstein, Eric W. *Fast Fourier Transform*, From *MathWorld*–A Wolfram Web Resource. http://mathworld.wolfram.com/FastFourierTransform.html (29.09.2009)
[3_09]	Cooley, James W., and John W. Tukey, 1965, *An algorithm for the machine calculation of complex Fourier series*, Math. Comput. **19**: 297–301, http://www.amath.washington.edu/~narc/win08/papers/cooley-tukey.pdf (29.09.2009)
[3_10]	Amara Graps: "An Introduction to Wavelets", *IEEE Computational Sciences and Engineering,* Volume 2, Number 2, summer 1995, pp 50–61, http://www.amara.com/ftpstuff/IEEEwavelet.pdf (29.09.2009)
[3_11]	Desanka Radunović: *Talasići za neupućene*, Petnica, 2005, http://poincare.matf.bg.ac.yu/~dradun//petnica.pdf (29.09.2009)
[3_12]	Subhasis Saha: Image Compression – from DCT to Wavelets : A Review, The ACM Student Magazine, http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html (21.08.2009)
[3_13]	Jason Elzinga and Keith Feenstra: *JPEG 2000: The Next Compression Standard using wavelet technology*, 2001, http://faculty.gvsu.edu/aboufade/web/wavelets/student_work/EF/ (21.08.2009)
[3_14]	Léon Bottou, Patrick Haffner, Paul G. Howard, Patrice Simard, Yoshua Bengio and Yann Le Cun: *High Quality Document Image Compression with DjVu, Journal of Electronic Imaging*, 7(3):410–425, 1998 http://leon.bottou.org/publications/pdf/jei-1998.pdf (20.08.09)
[3_15]	DjVu.org : *What is DjVu - DjVu.org*, http://djvu.org/resources/whatisdjvu.php. (16.10.2009)
[3_16]	John Kominek: *Introduction to Fractal compression,* Search the FAQ Archives, **http://www.faqs.org/faqs/compression-faq/part2/section-8.html** (4.10.2009)
[3_17]	Tal Kubo: *What is the state of fractal compression?*, Search the FAQ Archives**, http://www.faqs.org/faqs/compression-faq/part1/section-15.html (4.10.2009)**

Poglavlje 4: **Digitalni vodeni žig**

[4_01] I J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, T. Kalker: *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, 2008

[4_02] Vesna Vučković: *Digital Watermark*, Pregled Nacionalnog centra za digitalizaciju, 5(2003), 59–62,

http://www.ncd.matf.bg.ac.yu/casopis/05/Vuckovic/Vuckovic.pdf (4.10.2009)

[4_03] Vesna Vučković: *Digitalni vodeni žig i njegova uloga u digitalizaciji kulturne baštine*, Pregled NCD 7 (2005), 8–13, http://www.ncd.matf.bg.ac.yu/casopis/07/008/NCD07008.pdf (15.10.2009)

[4_04] F. C. Mintzer, L. E. Boyle, A. N. Cazes, B. S. Christian, S. C. Cox, F. P. Giordano, H. M. Gladney, J. C. Lee, M. L. Kelmanson, A. C. Lirani, K. A. Magerlein, A. M. B. Pavani, F. Schiattarella: *Toward on–line, worldwide access to Vatican Library materials,* IBM Journal of Research and Developement, Volume 40, Number 2, 1996, http://researchweb.watson.ibm.com/journal/rd/402/mintzer.html (8.08.2005)

[4_05] Fred Mintzer, Jeffrey Lotspiech, Norishige Morimoto: *Safeguarding Digital Library Contents and Users – Digital Watermarkin*g, IBM Research Division, D–Lib Magazine, December 1997, http://www.dlib.org/dlib/december97/ibm/12lotspiech.html (16.08.2009.)

Poglavlje 5: **Praksa žiga, steganografija, kriptografija**

[5_01] Neil F. Johnson: *Steganography*, Technical Report. November 1995, http://www.jjtc.com/stegdoc/steg1995.html (4.10.2009)

[5_02] Neil F. Johnson, Sushil Jojodia: *Exploring Steganograhy: Seeing the Unseen*, IEEE Computer, 1998, 26–34, http://www.jjtc.com/pub/r2026.pdf (15.10.2009)

[5_03] Francesco Queirolo: Steganography in Images – Final Communications Report, http://eric.purpletree.org/file/Steganography%20In%20Images.pdf

[5_04] Stefan Katzenbeisser, Fabien A. P. Petitcolas: *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Books, January 2000 (Chapter 3: Neil F. Johnson, Stefan C. Katzenbeisser: *A survey of steganographic techniques,* http://www.artechhouse.com/GetBlob.aspx?strName=Petitcolas035-ch03.pdf (15.10.2009)

[5_05] Mars Van Droogenbroeck, Jerome Delvaux: *An Entropy Based Technique for Information Embedding in Images*, Proc. 3rd IEEE Benelux Signal Processing Symposion (SPS–2002), Leuven, Belgium, 21–22, 2002, http://www.ulg.ac.be/telecom/publi/publications/mvd/sps-034.pdf, (4.10.2009)

[5_06] Peter Wayner: *Mimic Functions,* http://www.nic.funet.fi/pub/crypt/old/mimic/mimic.text (15.10.2009)

Poglavlje 6: **Svojstva žiga**

[6_01] M. Kutter, F.A.P. Petitcolas: A Fair Benchmark for Image Watermarking Systems, Electronics Imaging '99 Security and Watermarking of Multimedia Contents, vol. 3657, 1999, http://www.petitcolas.net/fabien/publications/ei99-benchmark.pdf (15.10.2009)

[6_02] Potdar, V.M. Han, S. Chang, E.: *A survey of digital image watermarking techniques***,** : 3rd IEEE International Conference INDIN '05. 2005, 709–716 http://debii.curtin.edu.au/~vidy/publications/INDIN_2005_A%20Survey%20of

%20Digital%20Image%20Watermarking%20Techniques.pdf (15.10.2009)

[6_03] Scott Craver , Nasir Memon , Boon–Lock Yeo , Minerva Yeung : *Can Invisible Watermarks Resolve Rightful Ownerships?*, IBM Research Technical Report RC 20509, July 1996)

[6_04] Craver, S. Memon, N. Yeo, B.–L. Yeung, M.M.: *Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications,* IEEE Journal on Selected Areas in Communications, May 1998, Volume: 16, Issue: 4, pages 573–586, http://www.cs.ucla.edu/~miodrag/cs259-security/craver98resolving.pdf (15.10.2009)

[6_05] [34] Craver, S.; Memon, N.; Boon–Lock Yeo; Yeung, M.M.: *On the invertibility of invisible watermarking techniques*, ICIP'97, Volume 1, 540–543

[6_06] Fabien a.P. Petitcolas, Ross J. Anderson, Markus G. Kuhn: *Attacks on Copyright Marking Systems*, Second workshop on information hiding, in vol. 1525 of Lecture Notes in Computer Science, 1998, pp. 218–238, http://www.petitcolas.net/fabien/publications/ih98-attacks.pdf (15.10.2009)

[6_07] S. Voloshynovskiy, S. Pereira, T. Pun: *Watermark attacks*, Erlangen Watermarking Workshop 99 October 5–6, 1999, http://vision.unige.ch/publications/postscript/99/VoloshynovskiyPereiraPun_eww99.pdf (1.10.2009)

[6_08] Fabien A. P. Petitcolas: Stirmark benchmark 4.0, http://www.petitcolas.net/fabien/watermarking/stirmark/ (15.10.2009)

[6_09] *Rec. ITU-R BT.500-11 - Methodology for the subjective assessment of the quality of television pictures*, http://www.dii.unisi.it/~menegaz/DoctoralSchool2004/papers/ITU-R_BT.500-11.pdf (17.10.2009)

Poglavlje 7: **Algoritmi digitalnog vodenog žiga**

[7_01] N. Nikolaidis, I. Pitas: *Robust image watermarking in the spatial domain*, Signal Processing 66 (1998) 385–403

[7_02] Phen–Lan Lin: Oblivious Digital Watermarking Scheme with Blob–Oriented and Modular–Arithmetic–Based Spatial–Domain Mechanism, Journal of Visual Communication and Image Representation 12, 136–151 (2001)

[7_03] Guo–Rui Feng, Ling–Ge Jiang, Dong–JianWang, Chen He: Quickly tracing detection for spread spectrum watermark based on effect estimation of the affine transform, Pattern Recognition 38 (2005) 2530 – 2536

[7_04] E. Koch, J. Zhao: *Towards Robust and Hidden Image Copyright Labeling*, IEEE Workshop on Nonlinear Signal and Image Processing, Jun. 1995, pp. 452–455

[7_05] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon: *Secure Spread Spectrum Watermarking for Images, Audio and Video*, Proceedings, International Conference on Image Processing, vol. III, pp. 243–246, (1996)

[7_06] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon: *A Secure, Robust Watermark for Multimedia*, Workshop on Information Hiding, Newton Institute, Univ. of

Cambridge, May 1996

[7_07] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon: *Secure Spread Spectrum Watermarking for Multimedia*, IEEE Trans. on Image Processing, 6, 12, 1673–1687, (1997)

[7_08] Mauro Barni, Franco Bartolini, Vito Cappellini, Alessandro Piva: *A DCT–domain system for robust image watermarking*, Signal Processing 66 (1998) 357–372

[7_09] Joachim J. Eggers, Bernd Girod: *Quantization effects on digital watermarks*, Signal Processing 81 (2001) 239–263

[7_10] M. L. Miller, G. J. Doerr, I. J. Cox: *Applying Informed Coding and Embedding to Design a Robust, High Capacity Watermark*, IEEE Trans. on Image Processing, 13, 6, 792–807, June 2004

[7_11] Miller, M.L. Dorr, G.J. Cox, I.J.: *Dirty–paper trellis codes for watermarking*, International Conference on Image Processing. 2002.

Volume: 2, 129–132

[7_12] Rongrong Ni, Qiuqi Ruan, H.D. Cheng: *Secure semi–blind watermarking based on iteration mapping and image features*, Pattern Recognition 38 (2005) 357 – 368

[7_13] Mohammad Eyadat, Shantaram Vasikarla: Performance evaluation of an incorporated DCT block–based watermarking algorithm with human visual system model, Pattern Recognition Letters 26 (2005) 1405–1411

[7_14] Fan Kefeng, Wang Meihuu, Mo Wei, Zhao Xinhua: *Novel copyright protection scheme for digital content*, Journal of Systems Engineering and Electronics, Vol. 17, No. 2, 2006, pp. 423–429

[7_15] Yi–Ta Wua, Frank Y. Shih: *Digital watermarking based on chaotic map and reference register*, Pattern Recognition 40 (2007) 3753 – 3763

[7_16] Wei Lu, Hongtao Lu, Fu–Lai Chung: *Novel robust image watermarking using difference correlation detector*, Computer Standards & Interfaces 29 (2007) 132–137

[7_17] Joseph J.K.O Ruanaidh, Gabriella Csurka, *A Bayesian Approach To Spread Spectrum Watermark Detection and Secure Copyright Protection for Digital Image Libraries*, cvpr, p. 1207, 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99) – Volume 1, 1999

[7_18] Xiaojun Qi, Ji Qi: *A robust content–based digital image watermarking scheme*, Signal Processing 87 (2007) 1264–1280

[7_19] Florent Autrusseau, Patrick Le Callet: A robust image watermarking technique based on quantization noise visibility thresholds, Signal Processing 87 (2007) 1363–1383

[7_20] Houng–Jyh Wang, C.–C.Jay Kuo: *Image Protection via Watermarking on Perceptually Significant Wavelet Coefficients*, IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing, December 7–9, 1998, Los Angeles, (279–284)

[7_21] Sanghyun Joo, Youngho Suh, Jaeho Shin, and Hisakazu Kikuchi: *A New*

Robust Watermark Embedding into Wavelet DC Components, ETRl Journal, Volume 24, Number 5, October 2002

[7_22] Zhao Dawei, Chen Guanrong, Liu Wenbo: *A chaos–based robust wavelet–domain watermarking algorithm*, Chaos, Solitons and Fractals 22 (2004) 47–54

[7_23] Jiang–Lung Liu, Der–Chyuan Lou, Ming–Chang Chang, Hao–Kuan Tso: *A robust watermarking scheme using self–reference image*, Computer Standards & Interfaces 28 (2006) 356– 367

[7_24] FrankY. Shih, Scott Y.T. Wu: *Combinational image watermarking in the spatial and frequency domains*, Pattern Recognition 36 (2003) 969 – 975

[7_25] Yulin Wang, Alan Pearmain: *Blind image data hiding based on self reference*, Pattern Recognition Letters 25 (2004) 1681–1689

[7_26] Xianyong Wu, Zhi–Hong Guan: *A novel digital watermark algorithm based on chaotic maps*, Physics Letters A 365 (2007) 403–406

[7_27] Huiyan Qi, Dong Zheng, Jiying Zhao: *Human visual system based adaptive digital image watermarking,* Signal Processing 88 (2008) 174–188

[7_28] Jonathan K. Su, Frank Hartung and Bernd Girod: *Digital watermarking of text, image, and video documents*, Comput. & Graphics, Vol. 22, No. 6, pp. 687–695, 1998

[7_29] Joseph J.K. O Ruanaidh, Thierry Pun: Rotation, scale and translation invariant spread spectrum digital image watermarking, Signal Processing 66 (1998) 303–317

[7_30] Severine Baudry, Philippe Nguyen, Henri Maitre: *Optimal decoding for watermarks subject to geometrical attacks*, Signal Processing: Image Communication 18 (2003) 297–307

[7_31] Guo–rui Feng, Ling–ge Jiang, Chen He, Yi Xue: *Chaotic spread spectrum watermark of optimal space–filling curves*, Chaos, Solitons and Fractals 27 (2006) 580–587

[7_32] Jake Richter: *The Digital Watermark*, (column in) PC Graphics Report, 1996, http://www.richterscale.org/pcgr/pc960220.htm (15.10.2009)

[7_32] M. Costa. *Writing on dirty paper*. IEEE Trans. Inform. Theory, 29:439–441, 1983.

[7_33] M. L. Miller, G. J. Doerr, I. J. Cox: *Applying Informed Coding and Embedding to Design a Robust, High Capacity Watermark*, IEEE Trans. on Image Procesing, 13, 6, 792–807, June 2004, http://www.ee.ucl.ac.uk/%7Eicox/papers/2004/TransIP03.pdf (15.10.2009)

[7_35] G. L. Miller, G. J. Doerr and I. J. Cox: *Dirty–Paper trellis codes for Watermarking*, IEEE Int. Conf. on Image Processing, II–129 – II–132 vol.2, 2002

[7_36] Lin Lin, Gwenaël Doërr, Ingemar J. Cox, Matt L. Miller: *An Efficient Algorithm for Informed Embedding of Dirty–Paper Trellis Codes for Watermarking*, IEEE Int. Conf. on Image Processing, 2005, http://www.ee.ucl.ac.uk/%7Eicox/papers/2005/ICIP2005.pdf

Poglavlje 8: **AWGN žig**

[8_01] Vesna Vučković: Embedding strength criteria for AWGN watermark, robust against expected distortion, Computing and Informatics, Vol. 29, no.3 (2010), p. 357–387

A. **Korisni linkovi**

[A_01] *Wikipedia, the free encyclopedia*, http://www.wikipedia.org/wiki/

[A_02] *Webopedia*, http://www.webopedia.com

[A_03] Watermarking World – Digital Watermarking Frequently Asked Questions (FAQ), http://www.watermarkingworld.org/faq.html (17.10.2009)

[A_04] MATLAB – The Language of Technical Computing (Matlab 6.5/13, Disc 2, *\help\pdf_doc\matlab\using_ml.pdf*)

[A_05] Image Processing Toolbox – For Use with MATLAB (Matlab 6.5/13, Disc 2, *\help\pdf_doc\images\images_tb.pdf*)

Sadržaj