

Matematičke metode u kompresiji slikovnog signala

Matošin, Lana

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:674894>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-02-24**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**MATEMATIČKE METODE U KOMPRESIJI SLIKOVNOG
SIGNALA**

Rijeka, rujan 2022.

Lana Matošin

0069076222

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**MATEMATIČKE METODE U KOMPRESIJI SLIKOVNOG
SIGNALA**

Mentor: Doc. dr. sc. Ivan Dražić

Komentor: Prof. dr. sc. Viktor Sučić

Rijeka, rujan 2022.

Lana Matošin

0069076222

Rijeka, 8. ožujka 2021.

Zavod: **Zavod za matematiku fiziku, strane jezike i kineziologiju**
Predmet: **Inženjerska matematika ET**
Grana: **1.01.07 primijenjena matematika i matematičko modeliranje**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Lana Matošin (0069076222)**
Studij: **Preddiplomski sveučilišni studij elektrotehnike**

Zadatak: **Matematičke metode u kompresiji slikovnog signala // Mathematical methods in image signal compression**

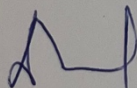
Opis zadatka:

U radu je potrebno opisati temeljne pristupe pri digitalizaciji i kompresiji slikovnog signala te dati pregled različitih formata datoteka u vezi s opisanim pristupima. Nadalje je potrebno ukratko objasniti elemente Fourierove analize i matičnog računa koji se koriste pri kompresiji slike. U završnom dijelu rada potrebno je provesti nekoliko eksperimenata kojima će se demonstrirati razlike opisanih metoda.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Lana Matošin
Zadatak uručen pristupniku: 15. ožujka 2021.

Mentor:



Doc. dr. sc. Ivan Dražić



Prof. dr. sc. Viktor Sučić (komentor)

Predsjednik povjerenstva za
završni ispit:



Prof. dr. sc. Viktor Sučić

ZAHVALA

IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 01. veljače 2020., izjavljujem da sam samostalno izradio/la završni rad prema zadatku preuzetog dana 15. ožujka, 2021.

Rijeka, rujan, 2022.

Lana Matošić

Lana Matošić

Ime i prezime

ZAHVALA

Prvenstveno veliko hvala mojim roditeljima, majci Jasni Dodić - Matošin i ocu Jurici Matošin, koji su oduvijek vjerovali u mene i moj uspjeh. Iako je bilo dosta izazova s kojima sam se susretala tijekom svog preddiplomskog školovanja znala sam da uz njih nikad nisam sama. Hvala vam na maksimalnom strpljenju i svojoj danoj ljubavi i bez sebičnom dijeljenju i ulaganju u mene i moj osobni razvoj.

Hvala mojim sestrama i bratu, bez čije podrške i ljubavi također ni jedan moj uspjeh ne bi bio moguć. Uvijek su uz mene i uz njih cijeli ovaj tijek mog studiranja prolazi puno lakše, zanimljivije i zabavnije.

Hvala i mom mentoru doc. dr. sc. Ivanu Dražiću na stalnoj dostupnosti, te svojoj pomoći i uloženom vremenu. Hvala Vam što ste našli vremena i imali strpljenja za sva moja pitanja.

SADRŽAJ:

1. UVOD	1
2. KOMPRESIJA BEZ GUBITAKA.....	2
2.1 Run – lenght kodiranja.....	2
2.2 Kodiranje područja	3
2.3 Huffmanovo kodiranje	3
2.4 Lempel – Ziv metoda	7
2.4.1 Lempel Ziv Welch metoda	8
3. KOMPRESIJA S GUBITCIMA	9
3.1 Transformacijsko kodiranje	9
3.2 Vektorka kvantizacija (VQ).....	10
3.3 Fraktalno kodiranje.....	11
3.4 Prediktivno kodiranje.....	11
4. JPEG.....	12
5. PNG.....	16
6. GIF.....	19
7. TIFF.....	22
8. BMP	24
9. USPOREDBA U ODNOSU NA KVALITETU I SNAGU KOMPRESIJE.....	25
10. ZAKLJUČAK.....	32
11. POPIS LITERATURE	33
SAŽETAK.....	36
SUMMARY.....	37
POPIS PRILOGA.....	38

1. UVOD

Tema ovog završnog rada je matematičke metode u kompresiji slikovnog signala. U vrijeme svjetske globalizacije i razvoja sve je veća potreba za prijenosom što veće količine podataka. Kompresijom podataka mogu se postići traženi i potrebni rezultati. Prijenos velikih slikovnih datoteka predstavlja usko grlo distribuiranih sustava

Svrha ovog završnog rada je da ukaže na važnost primjene i upotrebe matematičkih metoda u kompresiji slikovnog signala. Kod kompresije vrlo je bitna kako prenosivost tako i performanca. Cilj rada je provesti istraživanje matematičkih metoda u kompresiji slikovnog signala, te analizirati i usporediti slikovne formate (JPEG, PNG, GIF, TIFF i BMP). Svaki od spomenutih formata se ustvari i zasniva na matematičkim zakonitostima.

Rad je podijeljen na deset poglavlja, od kojih je prvo poglavlje uvod u kojem se definira tema, svrha i cilj rada te metode koje su korištene pri pisanju ovog završnog rada.

U drugom poglavlju se govori o kompresiji bez gubitka, odnosno pojašnjavaju se metode kodiranja Run – length i Huffmanovo kodiranje, te Lemel Ziv metoda. Treće poglavlje se bavi kompresijom sa gubitkom, te transformacijskim kodiranjem, vektorskom kvantizacijom, fraktalnim i prediktivnim kodiranjem. Metoda kompresije s gubitkom i metoda kompresije bez gubitka razlikuju se u prirodi ljudskog osjetila vida, gdje se njegove osobine koriste kod kompresije s gubitkom (dozvoljava određeni gubitak izvornih podataka) dok kod kompresije bez gubitaka kompresirani podatci su sačuvani i sliku je moguće u potpunosti rekonstruirati nakon dekompresije.

Kroz sljedećih pet poglavlja opisani su neki od primjera standardiziranih formata za kompresiju slike kao što su JPEG, PNG, GIF, TIFF i BMP kojima se upotrebljavaju gore spomenute metode. U devetom poglavlju opisna je usporedba slikovnih formata u odnosu na kvalitetu i snagu kompresije.

Pri pisanju ovog završnog rada korištene su metode indukcije i dedukcije, metoda analize, te klasifikacije

2. KOMPRESIJA BEZ GUBITAKA

Tehnike sažimanja bez gubitaka, kao što samo ime govori, ne uključuju gubitak informacija. Ukoliko su podaci komprimirani bez gubitaka, izvorni podaci mogu se u potpunosti oporaviti. Odnosno slika je identična kao original nakon dekompresije, svaki piksel ima istu vrijednost kao prije kompresije.

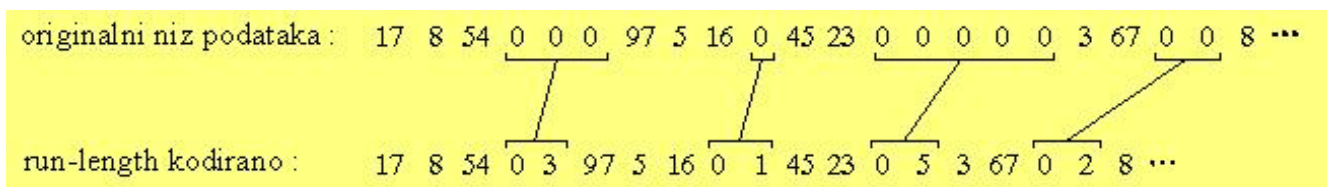
Metode koje omogućuju istovjetnost dekomprimirane slike i originalne su:

- Run-length kodiranje,
- Huffman kodiranje,
- Entropijsko kodiranje,
- Kodiranje područja [1].

2.1 Run – lenght kodiranja

Jednostavan oblik sažimanja podataka bez gubitaka u kojem se nizovi podataka pohranjuju kao jedna vrijednost podataka tj. nizovi podataka koji su u mnogim datotekama višestruko pojavljuju zamjenjuju se specijalnim znakovima, tzv. token [2]. Na taj način se smanjuje količina memorije koja je potrebna za pohranjivanje.

Najučinkovitija je za podatke koji sadrže mnogo takvih ponavljanja kao što su jednostavne grafičke slike, crteži linija i animacije, a nedostatak ove metode su ograničene mogućnosti kompresije.



Slika 2.1 Primjer Run – lenght kodiranja [3]

2.2 Kodiranje područja

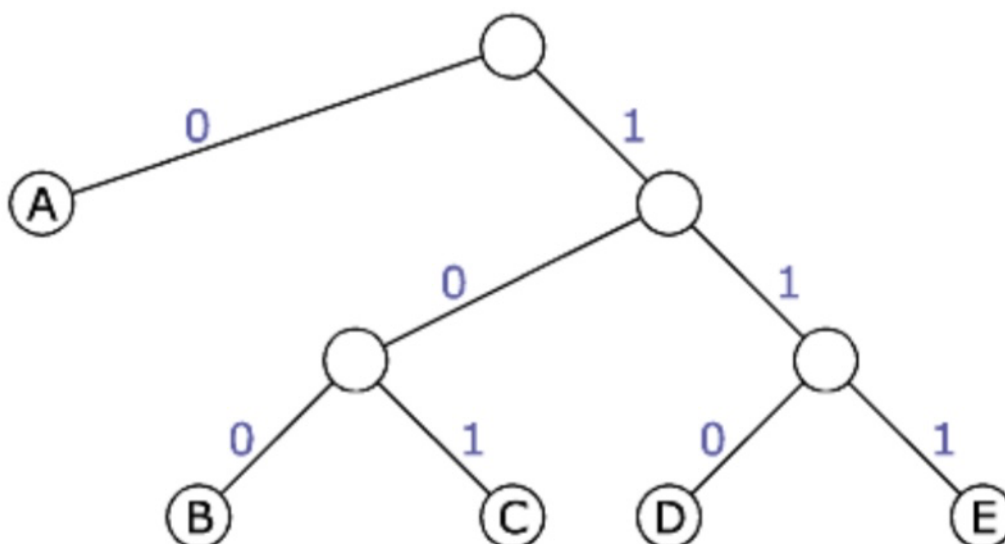
Kodiranje područja je poboljšana metoda run-length kodiranja, a princip kodiranja temelji se na iskorištavanju dvodimenzionalnih karakteristika slike. Koristi se ukoliko je određeni dio slike važniji od drugog, npr. neki objekt koji je u prvom planu i pozadina. Logično je da u većini slučajeva pozadina se smatra manje važnom od onoga što je u prvom planu.

Osnovni zadatak ovog tipa kodiranja je da izdvoji objekt u prvom planu i označi ga bitnim, a samim time i kodira kvalitetom bez gubitaka u odnosu na pozadinu [4]. Vrlo je djelotvoran način kodiranja, ali zbog nelinearnosti javlja se problem sklopovske implementacije pa je iz tih razloga relativno spor.

2.3 Huffmanovo kodiranje

Algoritam je razvio 1952.g. David Huffman za tekstualne datoteke. Princip rada baziran je na vjerojatnosti ponavljanja određenih znakova. Po tom principu algoritam izrađuje težinsko binarno stablo na osnovi frekvencije ponavljanja pojedinih znakova. Zatim se svakom pojedinom elementu stabla dodjeljuje novi kod koji je određen lokacijom znaka u težinskom binarnom stablu. Najčešći znak koji se ponavlja nazivamo korijen i on predstavlja najkraću kodnu riječ, a najrjeđe ponavljano znaku kodna riječ može biti dvostruko duža nego znak [5].

Prije pristupanja samom procesu kodiranja potrebno je napraviti statičku analizu slike, a zatim se izračuna vjerojatnost ponavljanja DCT koeficijenata. Slijedeći korak je izračun minimalnog broja bitova za svaki kvantizirani DCT koeficijent koji je različit od 0. Oni koeficijenti koji se učestalije ponavljaju imaju kraće kodne riječi i takvi koeficijenti nose manje značajne informacije, dok koeficijenti koji se rijetko pojavljuju imaju duže kodne riječi. Takvim pristupanjem se svakom koeficijentu pridružuje odgovarajući kod koji se bilježi u Huffmanovoj tablici. Kako bi dekodiranje bilo ispravno, dekodirer mora sadržavati identičnu tablicu. Kako bi se bolje shvatio princip Huffmanovog kodiranja potrebno je objasniti tzv. stablo [6].



Slika 2.2 Huffmanovo stablo[5]

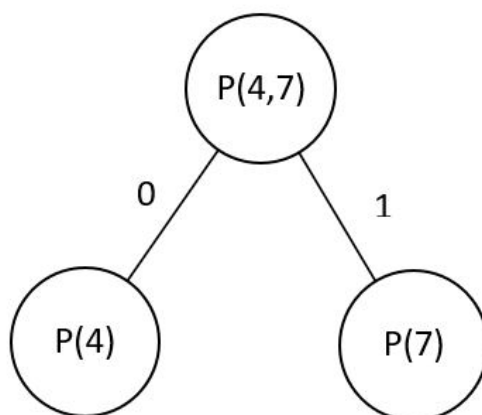
Na vrhu stabla nalazi se korijen stabla koji se dijeli na maksimalno dva čvora koja se još nazivaju djeca. Svaki slijedeći čvor se također dijeli na maksimalno dvoje djece. Stablo se kreće popunjavati od djece prema korijenu slijedeći Huffmanov algoritam [7].

Huffmanov algoritam:

1. Posložimo vjerojatnosti ponavljanja nekog znaka od najveće prema najmanjoj, tako da se najveća vjerojatnost nalazi na krajnoj lijevoj strani, a najmanja na krajnoj desnoj strani.
2. Zbrojimo dvije krajnje desne vjerojatnosti.
3. Zbrojimo dvije najmanje vjerojatnosti. U slučaju postojanja više jednakih najmanjih vjerojatnosti, zbrojimo bilo koje dvije, obično se zbrajaju dvije s krajnje desne strane.
4. Ponavljamo korak 3 dok ne ostane samo jedna vjerojatnost. Ako je stablo ispravno, ona će uvijek biti jednaka 1.
5. Počevši od vrha prema dnu, na svakoj točki na kojoj se stablo grana, dodijelimo po jednu binarnu vrijednost (0 ili 1) svakoj grani. Obično se lijevoj grani pridodaje 0 a desnoj 1. Ako se napravi obrnuto, kod će izgledati malo drugačije, ali će imati identična svojstva [6].

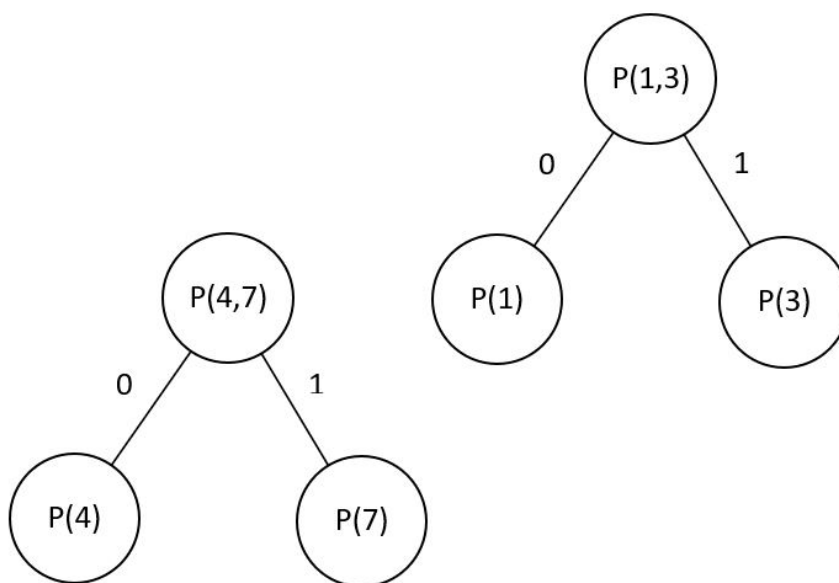
Na slijedećem primjeru objašnjen je princip rada Huffmanovog algoritma. Kako bi se pojednostavila situacija uzet je sljedeći niz podataka: 1 1 2 2 2 2 3 3 4 5 5 6 6 6 7 8 8. Zatim su izračunate vjerojatnosti pojavljivanja pojedinog znaka: $p(1) = 0.1176$, $p(2) = 0.1764$, $p(3) = 0.1176$, $p(4) = 0.0588$, $p(5) = 0.1176$, $p(6) = 0.1764$, $p(7) = 0.0588$, $p(8) = 0.1176$.

Slijedeći algoritam potrebno je uzeti dva znaka s najmanjim vjerojatnostima i oni će dati svog roditelja. U ovom primjeru uzeti su znakovi 4 i 7. Vjerojatnost njihovih roditelja bit će jednaka zbroju njihovim pojedinačnih vjerojatnosti odnosno u ovom slučaju iznosi 0.2352.

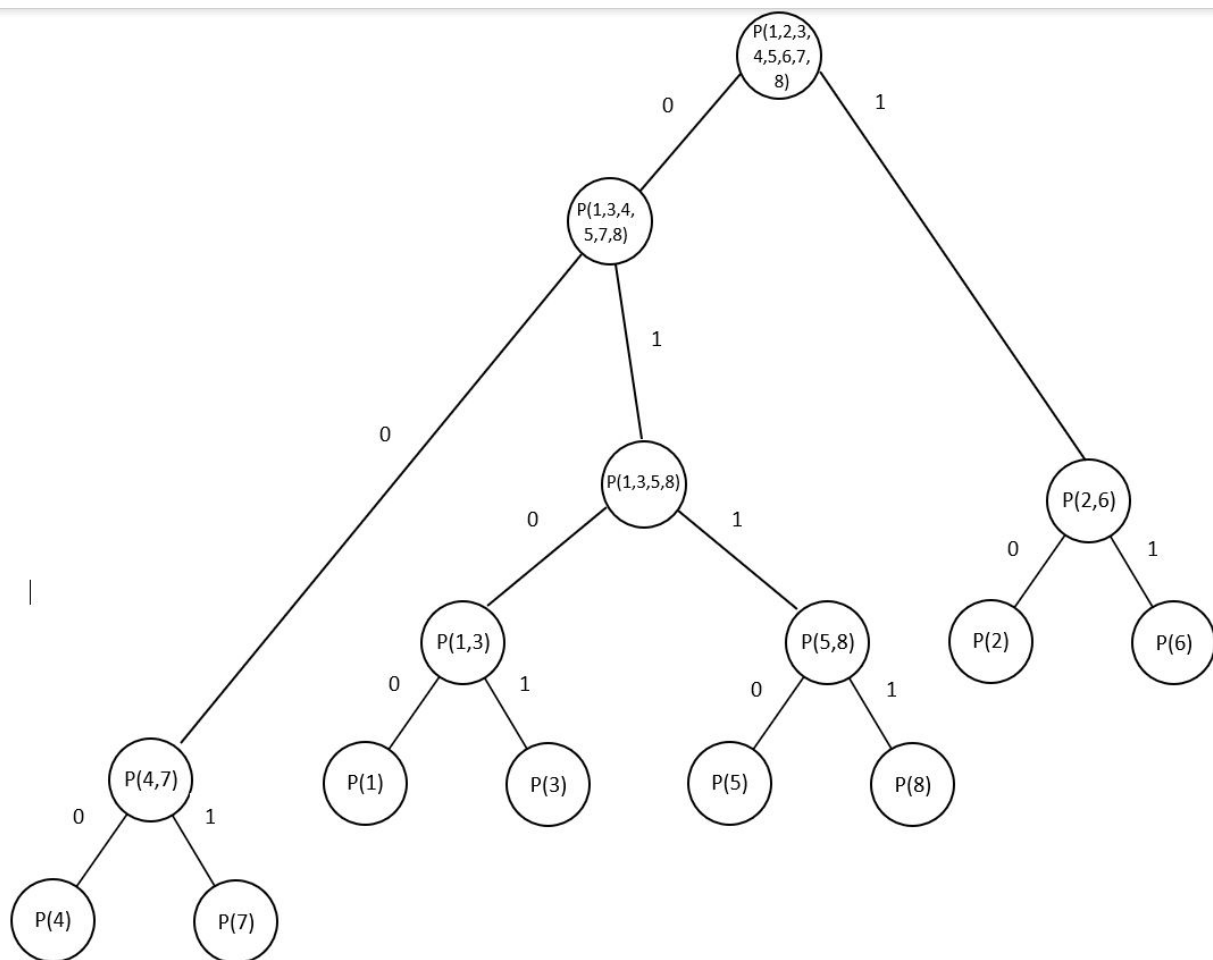


Slika 2.3 Primjer Huffmanovog stabla – I korak [6]

Postupak se ponavlja, slijedeći znakovi s najmanjim vjerojatnostima su 1 i 3.



Slika 2.4 Primjer Huffmanovog stabla – II korak [6]



Slika 2.5 Primjer Huffmanovog stabla - posljednji korak [6]

Tablica 2.1 Huffmanova tablica [6]

ZNAK	VJEROJATNOST	HUFFMANOV KOD
1	0.1176	0100
2	0.1764	10
3	0.1176	0101
4	0.0588	000
5	0.1176	0110
6	0.1764	11
7	0.0588	001
8	0.1176	0111

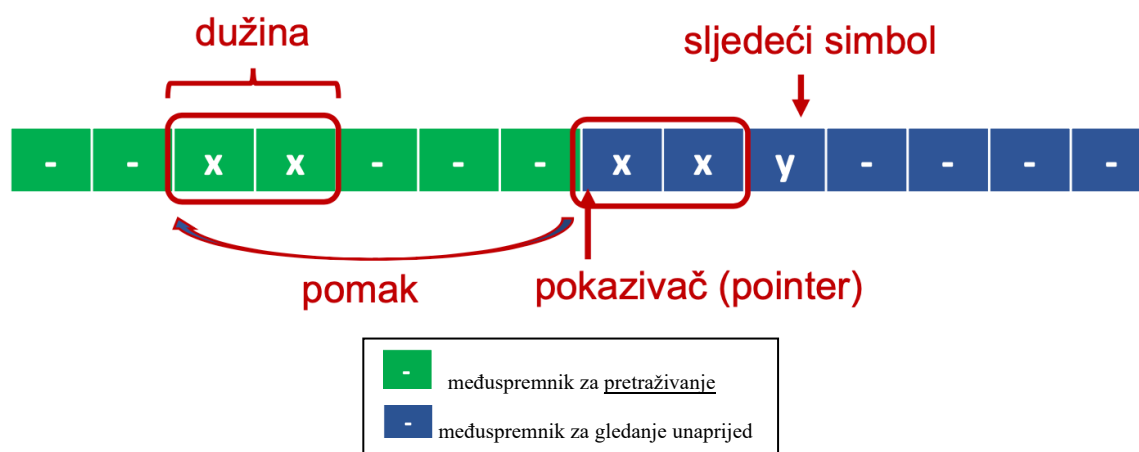
2.4 Lempel – Ziv metoda

Lempel- Ziv algoritam razvili su 1977. godine znanstvenici Abraham Lempel i Jacob Ziv. Algoritam koristi kompresiju podataka bez gubitaka. To nije jedan algoritam već cijela „obitelj“ algoritama, koja proizlazi iz dvaju algoritama. Lempel – Ziv algoritmi imaju široku primjenu u alatima za kompresiju kao što su gzip i GIF kompresija slike [5].

Prilikom kodiranja LZ77 metodom, koder analizira ulazne podatke pomoću tzv. kliznog prozora. Klizni prozor se sastoji od dva dijela:

- (1) međuspremnik za pretraživanje (search buffer) i
- (2) međuspremnik za gledanje unaprijed (look a head buffer) [8].

Međuspremnik za pretraživanje sadrži podatke koji su već kodirani, dok međuspremnik za gledanje unaprijed sadrži podatke koji još nisu kodirani. Ulazni dio niza kreće se kroz klizni prozor s lijeva na desno. Kako bi koder kodirao dio niza koji se nalazi u međuspremniku za gledanje unaprijed, pokazivač se pomiče unatrag kroz međuspremnik za pretraživanje sve dok se ne nađe na podudaranje sa spremnikom za gledanje unaprijed. Identificirani niz podudaranja se kodira uređenom trojkom (pomak, dulžina, sljedeći znak). Pomak označava poziciju najdužeg pronađenog niza u međuspremniku za pretraživanje u odnosu na pokazivač, dužina označava duljinu pronađenog podudaranja, a sljedeći znak označava simbol koji se nalazi u međuspremniku za gledanje unaprijed neposredno nakon pronađenog niza podudaranja [9].



Slika 2.6 Princip LZ77 metode [10]

2.4.1 Lempel Ziv Welch metoda

Lempel Ziv Welch algoritam kodiranja unaprijeđen je LZ78 algoritam. Ova poboljšana verzija je najpoznatija modifikacija i ponekad se pogrešno naziva Lempel Ziv algoritam. Objavio ga je Terry Welch 1984. godine. LZW također koristi riječnik kako bi zamijenio veće nizove podatak sa indeksima i tako postigao efekt kompresije [11].

Njegova velika prednost pred ostalim algoritmima koji se temelje na riječnicima, je to da za njegovu dekompresiju nije potrebno prenositi riječnik, već je algoritam tako napravljen da se riječnik može konstruirati iz kodiranih podataka, a to pridonosi značajnom povećanju stupnja kompresije [12].

Početni zapisi riječnika se inicijaliziraju pomoću 8-bitne ASCII tablice koja je jednoznačno definirana, tako da i koder i dekodeer kreću od istog početnog riječnika i time eliminiraju potrebu prenošenja rječnika [13]. Primjer ovakvog načina kodiranja detaljnije je objašnjen u poglavlju 6.

3. KOMPRESIJA S GUBITCIMA

Tehnike sažimanja s gubitkom kako i samo ime kaže uključuju gubitak informacija, a podaci koji su komprimirani pomoću tehnika s gubitkom općenito se nakon dekompresije ne mogu vratiti na izvornu kvalitetu. Primjena kompresije s gubitcima najčešća je kod multimedijalnog sadržaja, a nama najpoznatiji formati kompresije s gubitcima su JPEG i MP3. Ovim načinom kompresije dobiva se datoteka manjeg formata, ali i manje kvalitete. Iz tog razloga ovaj način kompresije je idealan za multimedijalne datoteke.

Princip ove kompresije temelji se na nedostacima ljudskog oka, a ono je osjetljivije na promjene u nijansama boje više u odnosu na promjene u osvjetljenju. Zbog toga se te dvije komponente pohranjuju u zasebne kanale. Dijelovi slike koji malo doprinose ukupnom dojmu više se komprimiraju. Kompresijom s gubitcima uzrokuje degradaciju slike, ali mogući su mnogo veći omjeri kompresije nego što je to moguće kompresijom bez gubitaka. JPEG format je standard za kompresiju slike s gubitcima, a metode koje omogućuju ovaj način kompresije su: transformacijsko kodiranje, skalarna i vektorska kvantizacija, fraktalno kodiranje i prediktivno kodiranje [1].

3.1 Transformacijsko kodiranje

Metoda koja je razvijena 70-ih godina i ima široku primjenu kroz JPEG normu. Sastoji se od transformacije signala iz prostorne u frekvencijsku domenu i kvantizacije. Ukoliko su valni oblici diskretni tada se vrši diskretna frekvencijska transformacija. Prije samog kodiranja poželjno je sliku podijeliti u manje blokove po 64 elementa slike, a zatim se primjenjuje diskretna transformacija na svaki pojedini blok. Blok se može predstaviti matricom A, a rezultat transformacije matricom B. Matrica B ne predstavlja amplitude elemenata slike pa se njeni elementi nazivaju transformacijski koeficijenti [14].

Zatim, transformacija provodi frekvencijsku analizu nad elementima bloka. Elementi u matrici B poredani su tako da se u gornjem lijevom kutu nalazi istosmjerni (DC) koeficijent koji predstavlja najveću vrijednost i nosi većinu energije elemenata slike u bloku informacija o slici i najbitniji je u rekonstrukciji iste. Preostali koeficijenti, u slučaju blokova po 8x8 piksela njih 63, nazivaju se izmjeničnim (AC) koeficijentima [14].

Ukoliko se AC koeficijenti nalaze u okolini DC koeficijenta odgovaraju nižim prostornim frekvencijama dok AC koeficijenti koji su smješteni u donjem desnom kutu matrice odgovaraju višim prostornim frekvencijama. Glavna zadaća transformacije je prijenos što je više moguće energije iz visokofrekvencijskih koeficijenata na istosmjerni koeficijent, a što manje na koeficijente koji ga okružuju. Sama transformacija je linearan proces pa nema gubitaka niti provođenja kompresije već njena uloga je da pripremi koeficijente na sljedeći korak, kvantizaciju [7].

Kvantizacijom se uklanjaju određeni visokofrekvencijski elementi pa je to postupak koji unosi gubitke. Princip kvantizacije zasnovan je na nesvršenosti ljudskog vizualnog sustava. Ljudsko oko puno je osjetljivije na promjene u nijansama boje nego na promjene u osvjetljenju. Sukladno tome ljudsko oko ne vidi visoke prostorne frekvencije stoga nema potrebe za danjim prenošenjem.

3.2 Vektorka kvantizacija (VQ)

Vektorska kvantizacija naziva se još i kvantizacija blokova ili kvantizacija podudaranja uzoraka. Ideju da kodiranje nizova izlaza ima veću prednost u odnosu na kodiranje pojedinačnih uzoraka prvi je razvio Shannon. Osnovna ideja je generirati reprezentativan skup nizova [15]. Izvorni izlazni niz koji je zadan predstavlja se jednim od elemenata reprezentativnog skupa. Izlaz izvora grupira se u blokove ili vektore. Na primjer; N uzastopnih uzoraka tretiraju se kao komponente N - dimenzionalnog vektora. Također, moguće je uzeti blok od N piksela sa slike i tretirati svaku vrijednost piksela kao komponentu vektora veličine ili dimenzije N .

Vektor izlaza izvora predstavlja ulaz u vektorski kvantizator. Skup N -dimenzionalnih vektora naziva se šifarnik (eng.codebook) i isti je za koder i dekoder. Vektori u šifarniku nazivaju se kodni vektori, a odabrani su tako da predstavljaju vektore generirane iz izlaza izvora. Svakom kodnom vektoru dodjeljuje se binarni indeks. U koderu se ulazni vektor uspoređuje sa svakim kodnim vektorom kako bi se pronašao kodni vektor koji je najbliži ulaznom. Elementi kodnog vektora su kvantizirane vrijednosti izlaza izvora. Kako bi dekoder prepoznao koji je kodni vektor najbliži ulaznom vektoru prenosi se ili pohranjuje binarni indeks kodnog vektora. Obzirom da dekoder ima isti šifarnik kao i koder može se dohvatiti vektor koda s obzirom na njegov binarni indeks [15].

3.3 Fraktalno kodiranje

Fraktalna kompresija metoda je temeljena na fraktalima. Fraktali su geometrijski objekti koji daju jednaku količinu detalja neovisno o razlučivosti. Odnosno, fraktal je moguće uvećavati beskonačno mnogo puta, a pri svakom novom uvećanju uočljivi su novi detalji. Približno su samoslični što bi značilo da su umanjena verzija samih sebe. Ova metoda kompresija najprikladnija je za slike prirode i teksture, a temelji se na činjenici da dijelovi slike nalikuju drugim dijelovima te iste slike [16].

Algoritmi pretvaraju te dijelove u matematičke podatke koji se nazivaju fraktalni kodovi. Nadalje, fraktalni kodovi se koriste za stvaranje kodirane slike. Ovom metodom zabilježeni su iznimni rezultati kao što je omjer kompresije 10000:1, ali nedostatak je što su algoritmi poprilično složeni i iziskuju previše vremena za izvođenje pa nisu primjenjivi u komercijalne svrhe.

3.4 Prediktivno kodiranje

Ovaj način kodiranja temeljen je na pretpostavci da postoji veza među susjednim uzorcima, odnosno da se signal neznatno promijeni između dva uzorka. Korelacija je moguća u vremenu, a također i u prostoru, a upotrebljava se da bi se predvidjele vrijednosti uzorka na temelju ranijih vrijednosti. Razlika između stvarnog i predviđenog uzorka se kodira, a niz tih razlika predstavlja signal razlike. Raspon vrijednosti signala razlike znatno je manji u odnosu na originalni signal. Zbog navedenog signal razlike se lakše kodira. Ovaj način kodiranja može se provoditi unutar slike na temelju ranije komprimiranih dijelova iste. Takvim postupanjem dolazi se do niza podataka koji je sažetiji i potrebno ga je kodirati. Iz tog razloga slika zauzima manje memorijskog prostora.

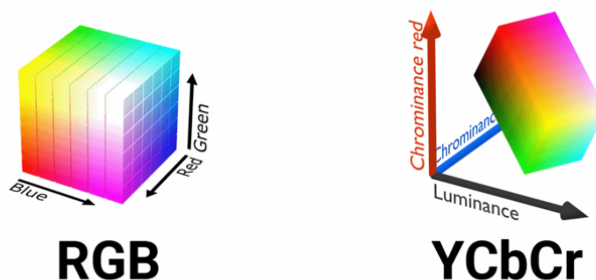
4. JPEG

Izraz JPEG je akronim za Joint Photographic Experts Group, koja je stvorila standard 1992. godine. JPEG je od svog uvođenja najrašireniji standard za kompresiju digitalne slike. Svaki dan od 2015. godine proizvede se nekoliko milijardi JPEG slika [17]. Također, JPEG je uvelike bio odgovoran za širenje digitalnih fotografija na internetu, a danas na društvenim mrežama. Stupanj kompresije je podesiv i korisnik može odabrati između veličine pohrane i kvalitete slike.

Osnovna metoda kompresije koju sadrži JPEG je metoda temeljena na diskretnoj kosinusovoj transformaciji koja pripada skupini kompresije s gubitcima. Postoje razni principi kompresije JPEG formata, ali uobičajeni princip se sastoji od transformacije prostora boja, podjele slike u blokove, diskretne kosinusne transformacije, kvantizacije i entropijskog kodiranja [17].

Računalne slike sastavljene su od mreže sitnih obojenih kvadrata, odnosno piksela, Svaki piksel je pohranjen kao tri broje, koji predstavljaju količinu crvene, zelene i plave svjetlosti potrebne za reprodukciju boje tog piksela. Zbog toga se i naziva RGB slika. Problem je u tome što se informacije o svjetlini slike ravnomjerno šire kroz R, G i B kanale; a ljudsko oko je puno osjetljivije na promjene u osvjetljenju nego na promjene u nijansama boje, zato je potrebno izolirati svjetlinu od informacija o boji.

RGB prostor preslikava se u YCbCr. YCbCr također ima tri kanala, ali sve informacije o svjetli pohranjuje u jedan zaseban kanal (Y), dok se informacije o boji dijele u preostala dva kanala (Cb i Cr). Nakon toga JPEG odbacuje neke informacije o boji utječući samo na kanale Cb i Cr dok najvažniji kanala Y zadržava punu veličinu [18].



Slika 4.1 Preslikavanje RGB prostora u YCbCr [19]

Iskorištavajući drugo zapažanje o ljudskoj vizualnoj percepciji, svaki od Y, Cb i Cr kanala dijeli se na 8x8 blokova piksela. U slučaju da ukupan broj elemenata slike u retku ili stupcu nije djeljiv s 8, potrebno je kopirati zadnji redak elemenata slike i krajnji desni stupac elemenata slike sve dok se ne dobije ukupan broj elemenata koji je djeljiv s 8.

Ključ procesa osnovne komprsije JPEG-a je matematička transformacija poznata kao diskretna kosinusna transformacija (DCT). DCT pripada skupini matematičkih operacija koje uključuju Fourierovu transformaciju. Osnovna svrha ovih operacije je uzeti određeni signal i transformirati ga iz jedne vrste u drugu vrstu reprezentacije.

Na primjer, DCT se može koristi za pretvaranje signala (prostorne informacije) u numeričke podatke (frekvencijske) tako da informacije o slici postoje u kvantitativnom obliku kako bi kompresija bila primjenjiva.

U nastavku su dane jednažbe koje opisuju matematički prikaz 8x8 DCT i 8x8 IDCT.

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (4.1)$$

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (4.2)$$

gdje je:

$$C(u), C(v) = \frac{1}{\sqrt{2}} \text{ za } u, v = 0$$

$$C(u), C(v) = 1 \text{ (inače)}$$

Slika 4.2 Matematički prikaz DCT [17]

DCT koeficijenti, $F(u,v)$, sadržavaju informaciju o frekvencijskom sastavu slike. Koeficijent $F(0,0)$ zove se istosmjerni ili DC koeficijent i nalazi se u lijevom gornjem kutu transformacijske matrice. Istosmjerni (DC) koeficijent predstavlja najviše informacija o slici, također najbitniji je u rekonstrukciji slike i nalazi se u gornjem lijevom kutu matrice. Preostali koeficijenti, u slučaju blokova po 8×8 piksela njih 63, nazivaju se izmjeničnim (AC) koeficijentima [20].

Ukoliko se AC koeficijenti nalaze u okolini DC koeficijenta odgovaraju nižim prostornim frekvencijama, dok AC koeficijenti koji su smješteni u donjem desnom kutu matrice odgovaraju višim prostornim frekvencijama. Prednost DCT-a je sklonost grupiranju većine signala u jedan kut rezultata [20].

Slijedeći korak je selektivno odbacivanje nekih informacija o frekvenciji odnosno kvantizacija. Kvantiziraju svi DCT koeficijenti na način da se koriste kvantizacijske tablice. Kvantizacijske matrice reduciraju amplitudu koeficijenata koji vrlo malo ili uopće ne utječu na kvalitetu slike. Cilj kvantizacije je povećanje nultih vrijednosti u matrici odnosno gubitak informacija koje nisu od prevelikog značaja.

Postupak kvantizacije se vrši tako da svaki DCT koeficijent dijeli pripadajućim elementom iz kvantizacijske tablice, a pri tom se koriste dvije kvantizacijske tablice. Jedna tablica je za informacije o svjetlini, a druga za informacije o boji. Dakle, gornji lijevi kut svakoog 8×8 bloka u Y frekvencijskom kanalu bit će podijeljen brojem u gornjem lijevom kutu kvantizacijske matrice za informacije o svjetlini, zatim se dobivena vrijednost zaokružuje na najbliži cijeli broj.

Kvantizirane koeficijente potrebno je poredati u "cik-cak" redoslijed kao što je prikazano na slici (neki broj). Takav redoslijed olakšava entropijsko kodiranje koje slijedi zato što brojevi koji su različiti od 0 (nisko frekvencijski koeficijenti) smješteni su na početku niza dok su nule (visoko-frekvencijski koeficijenti) smještene iza njih. Nakon toga se povodi Run Length kodiranje kako je opisano u poglavlju 2.1.

5. PNG

PNG (Portable Network Graphics) je bitmap slikovni format koji primjenjuje kompresiju bez gubitaka. Zamišljen je kao grafički format za razmjenu preko interneta, a ne za osobnu upotrebu. Prvi put se javlja 1995. godine kao zamjena za GIF format. Potreba za zamjenom GIF formata javlja se iz razloga što je Unisys najavio da će softverskim patentom zaštititi LZW kompresijski algoritam koji se koristi u kompresiji GIF formata [21].

GIF format je ograničen na 256 boja što je već za to vrijeme ograničavajuće jer su postojala računala koja su bila u mogućnosti prikazivati puno više. PNG format doživljava sve veću upotrebu kolovozu 1999. godine kada je Unisys ukinuo patentne dozvole, a 10.11.2003. postaje međunarodni standard (ISO/IEC 15948:2003) [21].

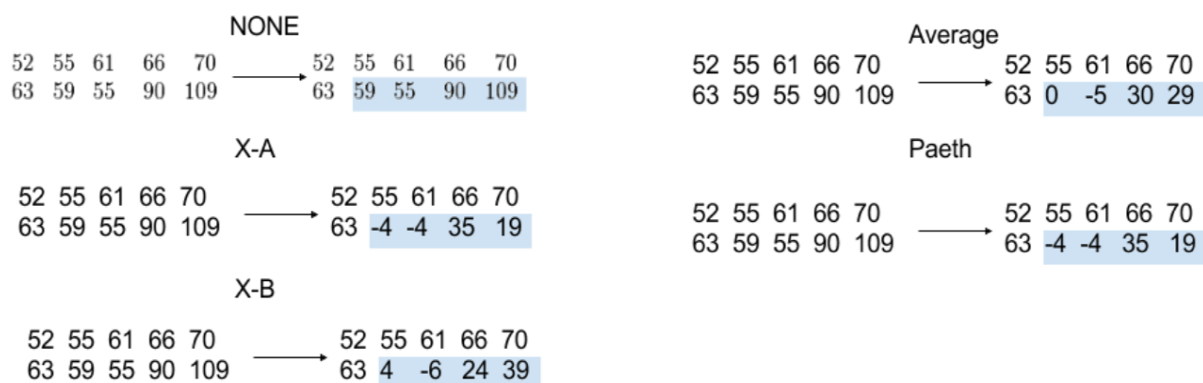
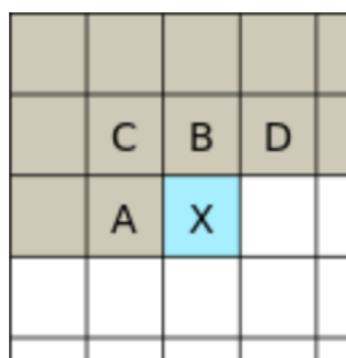
Potrebno se osvrnuti na mehanizam za kompresiju. PNG podržava korak predkompresije koji se naziva filtriranje. Filtriranje je metoda reverzibilne transformacije slikovnih podataka kako bi glavni mehanizam kompresije radio što učinkovitije. Za primjer može se razmotriti niz bajtova koji se ravnomjerno povećavaju od 1 do 255 [22].

U nizu nema ponavljanja stoga je sažimanje vrlo slabo ili uopće nije moguće, ali ukoliko se prvi bajt ostavi sam, a svaki slijedeći bajt se zamjeni razlikom između njega i prethodnika dobiva se niz koji je izuzetno kompresibilan odnosno skup od 255 identičnih brojeva od kojih svaki ima vrijednost 1. PNG podržava pet vrsta filtara, a koder može odabrati korištenje drugog filtra za svaki red piksela na slici [22].

Tablica 5.1 opisuje pet vrsta filtara koji se koriste kao korak predkompresije.

Tablica 5.1 Metode filtriranja [23]

IME	OPIS
None	Svaki byte ostaje nepromjenjen.
Sub	Svaki byte je zamjenjen razlikom između njega i prethodnika s lijeve strane.
Up	Svaki byte je zamjenjen razlikom između njega i byte iznad njega.
Average	Svaki bajt se zamjenjuje razlikom između njega i prosjeka odgovarajućih bajtova lijevo i iznad njega.
Paeth	Svaki bajt se zamjenjuje razlikom između njega i Paeth prediktora odgovarajućih bajtova lijevo, iznad njega i gore lijevo.



Slika 5.1 Primjeri metoda filtriranja [23]

Posljednju metodu filtriranja „Paeth“ potrebno je malo pobliže objasniti. Naime, metodu je izumio Alan Paeth, Paethov prediktor se izračunava na način da se prvo izračuna osnovna vrijednost koja je jednaka zbroju odgovarajućih bajtova lijevo i iznad, minus bajt koji se nalazi gore lijevo. Nakon toga se izračuna razlika između osnovne vrijednosti i svakog od tri odgovarajuća byte-a, a byte koji je dao najmanju apsolutnu razliku tj. najbliži je osnovnoj vrijednosti, koristi se kao prediktor i oduzima se od ciljanog byte-a kako bi se dobila filtrirana vrijednost.

Postavlja se pitanje kako odabrati najoptimalniji filter za svaki red. Programeri su eksperimentiranjem došli do nekih zaključaka da za crno bijele slike i slike u paletama upotrebljava se filter „None“, a ukoliko želimo koristiti isti filter za svaki red najbolje je upotrijebiti „Paeth“ [23].

Za crno-bijele slike i slike u bojama od 8 ili više bitova po uzorku pristupa se standardom minimalnog zbroja apsolutnih razlika. U takvom pristupu filtrirani byte-ovi tretiraju se kao vrijednosti s predznakom minus. Može se reći da se vrijednosti zrcale, odnosno svaka vrijednost iznad 127 se tretira kao negativna; 128 postaje -128, a 255 postaje -1. Potrebno je uzeti apsolutne vrijednosti svakog byte-a i zbrojiti sve zajedno za dati redak, a zatim usporediti zbrojeve za druge vrste filtera. Na posljertku se odabire filter koji daje najmanji zbroj.

Slijedeći korak je primjena DEFLATE kompresijskog algoritma koji objedinjuje LZ77 algoritam i Huffmanovo kodiranje. Kreće se od pretpostavke da se mnoge vrste podatka ponavljaju u različitim stupnjevima. Algoritam pronalazi duplicirane nizove u ulaznim podacima. Drugo pojavljivanje istog niza zamjenjuje se „pokazivačem“ na prethodni niz, u obliku para (udaljenost, duljina). Udaljenosti su ograničene na 32K byte-ova, a duljine na 258 byte-ova. Na kraju se duljine podudaranja komprimiraju se s jednim Huffmanovim stablom, a udaljenosti podudaranja komprimiraju se s drugim stablom [5].

6. GIF

Tvrtka CompuServe je 1987. godine predstavila GIF (Graphics Interchange Format) pod originalnim imenom 87a. GIF je nadomjestio raniji format RLE koji se također koristio za preuzimanje slika, ali je podržavao samo crno-bijele slike. 1989. godine ista tvrtka je napravila noviju poboljšanu verziju pod nazivom 89a još je poznata kao animirani GIF, odnosno slika koja je sastavljena od više „frameova“, koji daju dojam kretanje slike [24].

GIF je svoju popularnost doživio zbog korištenja Lempel-Ziv metode kodiranja koja je bila učinkovitija od Run-length metode pa su se poprilično velike slike mogle preuzeti relativno brzo i na sporim modemima. GIF je sveprisutan i danas na društvenim mrežama pa je tako u svibnju 2015. Facebook dodao podršku kao i Instagram u siječnju 2018 [24].

GIF slike komprimiraju se korištenjem Lempel-Ziv-Welch (LZW) metode kompresije podataka bez gubitaka. Kako bi primijenili LZW algoritam potrebno je svaki red piksela sagledati zasebno nakon čega se dobije preoblikovani oblik. Odnosno svi redovi piksela prikazani su u jednom redu [25].

Slijedeći korak je dodijeliti svakoj boji jedinstveni broj, a nakon što je svakoj boji dodijeljen jedinstveni broj, dobije se niz brojeva koji je potrebo komprimirati. Kako bi se primijenila LZW metoda, potreba je tablica nizova. Tablica nizova omogućit će upotrebu posebnih kodova za označavanje niza boja, a ne samo jednu po jednu [25].

Incijalizirana tablica nizova definirana je na temelju proizvoljne pretpostavke da slika sadrži tri boje, crvene bijele i zelene. Zatim primjenom koraka koji su navedeni u prošlom paragrafu dobiven je slijedeći niz brojeva 0 1 0 1 1 0 0 0 2 2 2 1.

Tablica 6.1 Inicijalizirana tablica nizova

KOD	BOJA
#0	0
#1	1
#2	2

Nakon definiranja inicijalizacijske tablice nizova može se pristupiti kompresiji podataka. Za početak potrebno je definirati „trenutni prefiks“ koji služi za pohranjivanje uspoređivanje piksela. U ovom slučaju označavat će se "[.c.]". Također, potrebno je definirati „trenutni niz“, koji predstavlja trenutni prefiks plus sljedeći znak u nizu. i označavat će se "[.c.]K", gdje je K neki znak.

Uzima se prvi znak u nizu, u ovom slučaju označen „P“. Trenutni niz je sada [.c.]P, pretraži se tablica kodova kako bi se utvrdilo pojavljuje li se [.c.]P. Obzirom da je [.c.]P već u tablici prelazi se na sljedeći korak. Sada neka [.c.]P bude trenutni prefiks, zatim se trenutnom prefiksu doda sljedeći znak u nizu, označen sa „Q“, pa se formira [.c.]Q trenutni niz.

Ponovno se pretražuje tablica nizova, ali u ovom slučaju [.c.]Q se ne pojavljuje u tablici. Potrebno je dodati u tablicu [.c.]Q (što je u ovom slučaju P Q) i označiti ga s #3 i ispiše se kod za [.c.]. Sada je potrebno krenuti ispočetka s trenutnim prefiksom koji je samo Q. Potrebno je dodavati znakove u [.c.] kako bi se formirao [.c.]K sve dok se ne može pronaći [.c.]K u tablici, a zatim se ispiše [.c.] i [.c.]K se dodaje u tablicu [26].

Kako bi gore naveden algoritam bio shvatljiviji pokazat će se na primjeru niza brojeva 0 1 0 1 1 0 0 0 2 2 2 1. Prvi znak u nizu je 0 i trenutni niz je [.c.]0. Pretraživanjem tablice, naravno, utvrđeno je da 0 nalazi u tablici. Zatim [.c.]0 postaje trenutni prefiks i uzima se sljedeći znak u nizu koji je 1 i trenutni niz postaje [.c.]1 odnosno 0 1. Ovaj slučaj nije u tablici nizova pa ga se označava sa #3 i dodaje u tablicu.

Nakon toga uzima se slijedeći znak u nizu koji je 0, nalazi se u tablici pa se dodaje slijedeći znak 1, koji je također u tablici pa trenutni prefiks postaje niz 0 1, kojem se dodaje slijedeći znak 1. Pretraživanjem tablice niz 0 1 1 se ne nalazi u istoj, stoga ga je potrebno dodati u tablicu i označiti sa #4.

Postupak se ponavlja sve dok se ne dođe do posljednjeg znaka u nizu, u ovom slučaju 1 i zato se na kraju ispisuje #1. Kodirani niz izgleda ovako: #0#1#3#4#0#0#2#8#1.

Tablica 6.2 Tablica kodova

KOD	BOJA
#0	0
#1	1
#2	2
#3	01
#4	10
#5	011
#6	100
#7	02
#8	22
#9	221

7. TIFF

TIFF (Tag Image File Format) format stvoren je 1986. godine od strane Aldus Corporation kao pokušaj da se proizvođači stolnih skenera usuglase oko zajedničkog formata datoteke skenirane slike. U početku je TIFF bio samo binarni format slike iz razloga što su tadašnji skeneri mogli samo takve obraditi. Napretkom skenera i povećanjem prostora za pohranu na disku stolnog računala napredovao je i TIFF. Prilagodio se slikama u boji i slikama u sivim tonovima, a danas je zajedno s JPEG i PNG vrlo popularan format za slike dubokih boja.

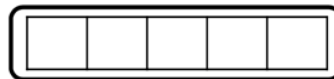
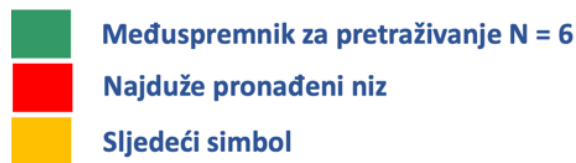
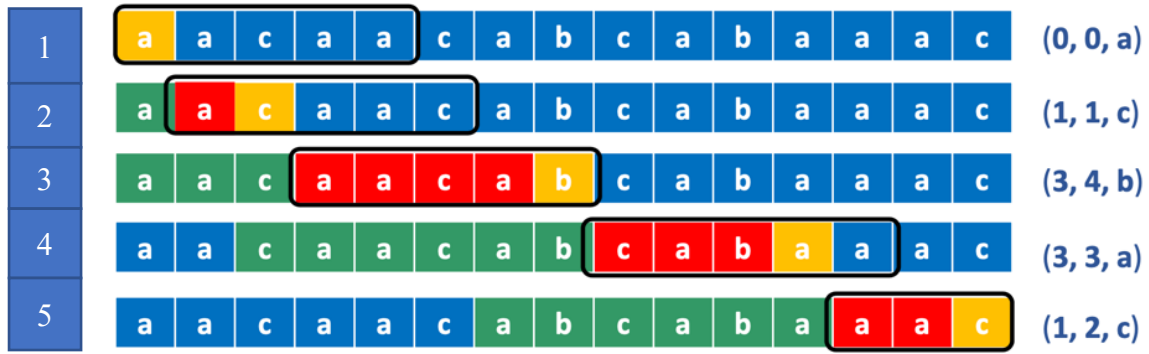
Kada je riječ o TIFF formatu moguće su dvije opcije kompresije:

1. Lampel-Ziv Welch (LZW) i
2. ZIP [27].

Obje ove metode su metode kompresije bez gubitaka. Ukoliko je potrebno komprimirati 8-bitnu datoteku obje metode kompresije će dati zadovoljavajuće rezultate, a kod 16-bitnih datoteka pogodnije je koristiti ZIP metodu. Obzirom da je LZW metoda kompresije podataka opisana kod GIF formata, u ovom poglavlju opisana je ZIP metoda.

Kao i prilikom kompresije PNG datoteke koristi se DEFLATE kompresijski algoritam koji se češće naziva Zip kompresija, a koristi kombinaciju LZ77 metode i Huffmanovog kodiranja. Smatralo se da algoritam koji proizvodi Deflate datoteke se može implementirati na način koji nije pokriven patentima. Upravo to je dovelo do njegove široke upotrebe – npr., u gzip komprimiranim datotekama i PNG slikovnim datotekama i uz ZIP format datoteke [5].

LZ77 algoritam koristi klizni prozor koji se pomiče zajedno s pokazivačem. Klizni prozor je podijeljen dva dijela: međuspremnik za pretraživanje i dio koji počinje od pokazivača – međuspremnik za gledanje unaprijed [28]. Osnovni algoritam je jednostavan i može se podijeliti u tri koraka. Prvi korak je pronaći najdulje podudaranje niza koji počinje od pokazivača i u potpunosti se nalazi u međuspremniku za pretraživanje sa nizom koji se nalazi u međuspremniku za gledanje unaprijed. Slijedeći korak je ispisati uređenu trojku (pomak, duljina, slijedeći simbol).. Posljednji korak je pomaknuti pokazivač na l+1 poziciju u desno. Navedeni koraci pokazat će se na primjeru niza „a a c a a c a b c a b a a a c“ .



Međusprennik za gledanje unaprijed L = 5

Slika 7.1 Primjer LZ77 algoritma [10]

Na početku međusprennik za pretraživanje je prazan i počinje se s lijeve strane, međusprennik za pretraživanje je uokviren, obzirom da nema podudaranja simbola iz međusprennika za gledanja unaprijed sa simbolima iz međusprennika za pretraživanje ispisuje se (0, 0, a). Žutom bojom označen je sljedeći simbol, a plavom ostatak niza.

U drugom koraku međusprennik za gledanje unaprijed se pomiče za jedno mjesto u desno, a međusprennik za pretraživanje označen je zelenom bojom, dok je crvenom bojom označeno najdulje podudaranje. Obzirom da je pronađeno podudaranje, pomak je 1, duljina podudaranja isto 1, a sljedeći simbol je c, ispisuje se (1, 1, c).

U trećem koraku međusprennik za gledanje unaprijed se pomiče iza simbola c i pronađeno je podudaranje tri znaka unatrag, a duljina niza podudaranja je četiri znaka, stoga se ispisuje (3,4,b). U četvrtom i petom koraku postupak se ponavlja, a na kraju se od početnog niza dobije kodirani niz u obliku uređenih trojki kako je prikazano na slici 7.1 (desno).

8. BMP

Microsoft je izvorno razvio BMP format za svoj operativni sustav Windows kako bi održao razlučivost digitalnih slika na različitim zaslonima i uređajima. BMP format još je poznat kao Device Independent Bitmap (DIB) format. Moguće je pohraniti dvodimenzionalne slike u boji ili crno – bijele. 1990-ih Microsoft je razvio BMP format datoteke kako bi omogućio prikaz slikovnih podataka neovisno o Windows uređajima. To je bilo iznimno bitno iz razloga što je novi BMP format, koji je spriječio da slike postanu zamućene ili izobličene, uklonio je potrebu za grafičkim adapterom [29].

BMP datoteka sadrži nekomprimirane podatke, što je čini idealnom za pohranu i prikaz digitalnih slika visoke kvalitete. S druge strane, nedostatak kompresije općenito stvara veće veličine datoteke od npr JPEG-a, PNG-a ili GIF-ova. BMP slike najčešće nisu komprimirane, ali u nekim uvjetima Run Length Encoding metoda kompresije je primjenjiva, također teoretski je moguće primijeniti JPEG, PNG i Huffmanov kompresijski algoritam, ali slučajevi koji to podržavaju su vrlo rijetki [30].

Glavna primjena BMP datoteke je za pohranjivanje visokokvalitetnih digitalnih fotografija. BMP datoteke dolaze u rasterskom formatu, što znači da su slike koje sadrže napravljene od mreže kvadratnih piksela, što omogućuje BMP-ovima pohranu slika sa širokim spektrom boja i detalja, a iz tih razloga su idealni za visokokvalitetne 2D digitalne fotografije. Obzirom da BMP-ovi nisu komprimirani mogu podnijeti visoke razine slikovnih podataka. Također, iz istih razloga zbog kojih su BMP datoteke izvrsne za pohranu fotografija, one su dobra opcija kod ispisa složenih fotografija u boji. Za razliku od JPEG-ova, koji komprimiraju podatke slikovne datoteke, BMP-ovi ne gube podatke svaki put kada ih uredite i spremite.

9. USPOREDBA U ODNOSU NA KVALITETU I SNAGU KOMPRESIJE

U ovom poglavlju rada napravljena je usporedba formata u odnosu na kvalitetu i snagu kompresije. Kako bi se napravila usporedba, izabrane su određene fotografije (priroda, La linea, Primošten, Crno – bijela fotografija). Navedene fotografije izvorno su u BMP formatu. U sljedećem koraku izabrane fotografije komprimirane su online („Convertio“ – www.Convertio.com) u ostale formate (JPEG, PNG, GIF I TIFF) čiji su principi kompresije detaljno obrađeni u prethodnim poglavljima ovog rada.

JPEG format u odnosu na BMP format i ostale zauzima najmanje memorijskog prostora, pogodan je za reljefne slike s puno prijelaza. Koristi se metoda kompresije s gubitcima pa zato datoteke zauzimaju malo memorijskog prostora. Upravo iz tog razloga je popularan na webu jer se slike lako i brzo učitavaju na društvene mreže i blogove pa je fotografije lako dijeliti s drugim korisnicima.

Velike brzine učitavanja prioritet su za današnje korisnike, osobito jer se posljednjih godina za pristup web mjestima koristi mobilni telefon umjesto stolnog računala. Također, ovaj format nailazi na široku primjenu kod spremanja fotografija u digitalnim fotoaparatom, a popularan je među fotografima.

Ovaj format nije pogodan ukoliko je potrebno nastaviti s brojnim izmjenama i ponovno spremati istu fotografiju. Svakim ponovnim spremanjem i uređivanjem gubi se na kvaliteti. Također, nije najbolji izbor prilikom ispisa slika, za slike koje sadrže tekst ili jednostavne oblike s oštrim rubovima. Prilikom kompresije tekst ili oštri detalji se mogu zamuti što će rezultirati fotografijom koja nije jasna i oštra kao što bi bila spremljena u drugom formatu.

PNG format sadrži puno više informacija u odnosu na JPEG i GIF zato će PNG slika zauzimati više memorijskog prostora. Upravo zato što sadrži više informacija ovaj format slike je puno kvalitetniji. Zajednička osobina s JPEG-om je prikladnost dijeljenja slika na webu. PNG datoteke za razliku od JPEG-a podržavaju transparentnost. Moguće je primijeniti prozirne pozadine na slike i podesiti stupanj transparentnosti.

Odnosno, PNG slike bolje integriraju s različitim bojama pozadine na stranici pa je samim time tekst lakši za čitanje. Ovaj format nailazi na široku primjenu kod prikazivanja raznih tablica, dijagrama, logotipa, karata, snimaka zaslona, statičkih slika i ostalih vizualnih elemenata koji uključuju tekst.

Obzirom da PNG format koristi kompresiju bez gubitaka treba ga izbjegavati ukoliko se radi o fotografijama visoke rezolucije i kod ispisa grafike profesionalne kvalitete.

GIF format je izvrsna vrsta datoteke za korištenje ako se radi s malom grafikom s ograničenom paletom boja, poput plakata, grafikona, tablica. Isto tako GIF slike idealne su za stvaranje kratke animacije. Kao i kod PNG-a, transparentnost je podržana u ovom formatu.

Mana GIF datoteka je ta što koriste paletu boja s indeksom od 256, što ograničava izbor boja pa fotografije mogu izgledati „ravno“ i bez sjaja, tj. ne može se postići zadovoljavajuća dubina bojama. GIF datoteke se spremaju u jednoslojnom formatu, što znači da se sva dodatna uređivanja spremaju u jedan sloj slike i ne mogu se vratiti. Prilikom ispisa slika, ovaj format je potrebno izbjegavati.

TIFF format često koriste profesionalci u kreativnim industrijama, takve datoteke zahtijevaju veliku količinu prostora za pohranjivanje pa je dijeljenje na webu otežano. TIFF datoteka odličan je odabir kada je cilj visoka kvaliteta, posebno kada je u pitanju ispis fotografija. Najbolji je za sve rasterske slike koje je potrebno uređivati i kada je potrebno sačuvati kvalitetu. Odnosno primjenjuje se na ilustracije, profesionalne fotografije, prilikom skeniranja dokumenata visoke kvalitete, ispisa slika visoke kvalitete, za slike u časopisima i novinama.

Zbog velike količine memorijskog prostora koji zauzima ova vrsta formata obično se ne koristi za svakodnevni marketing, odnosno TIFF datoteka nije pogodna kod malih datoteka ili formata koji su prilagođeni webu.



Slika 9.1 Priroda, (BMP, 492 KB (491.521 bytes)), La linea (16 KB (14.188 bytes)) [31,32]



Slika 9.2 Priroda (PNG, 170 KB (169.709 bytes)), La linea (5 KB (5.488 bytes)) [31,32]



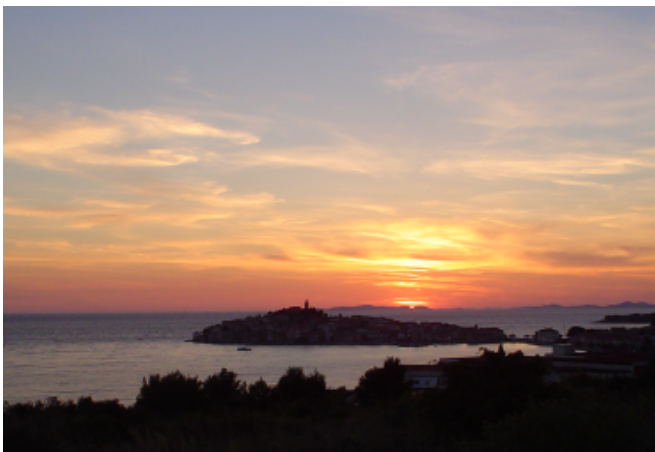
Slika 9.3 Priroda (GIF, 209 KB (208.617 bytes), La linea (GIF, 9 KB (8.651 bytes)) [31,32]



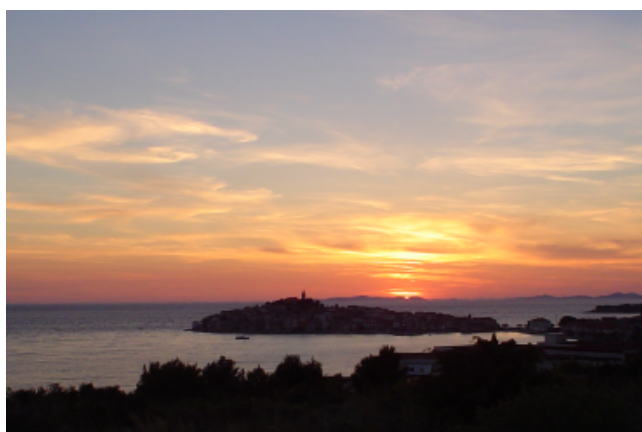
Slika 11 Priroda (TIFF, 487 KB (487.404 bytes)), La linea (TIFF, 51 KB (50.884 bytes)) [31,32]



Slika 9.5 Priroda,(JPEG, 71 KB (71.087 bytes)), La linea (JPEG, 8 KB (7.833 bytes)) [31,32]



Slika13 Primošten (BMP, 283 KB (280.470 bytes)), Crno - bijela fotografija (389 KB (387.156 bytes)) [31,32]



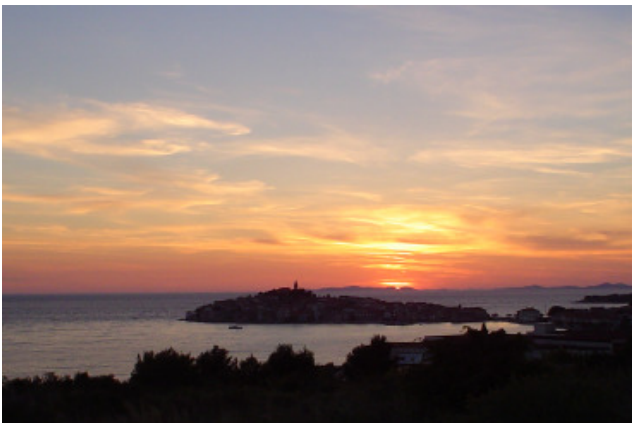
Slika 9.7 Primošten (PNG, 117 KB (116.839 byets)), Crno - bijela fotografija (78KB (76.169 bytes)) [31,32]



Slika15 Primošten (GIF, 54KB (53.888 bytes)), Crno - bijela fotografija (GIF, 110KB (110.116 bytes)) [33,34]



Slika16 Primošten (TIFF, 281 KB, 280.732 bytes), Crno - bijela fotografija (TIFF, 387KB (387.402 bytes)) [33,34]



Slika17 Primošten (JPEG, 24KB (24.018 bytes)), Crno - bijela fotografija (JPEG, 1,2 MB (1.150.637 bytes)) [33,34]

10. ZAKLJUČAK

U radu su detaljno opisane metode kompresije prilikom pohranjivanja u različite slikovne formate. Kroz slike prirode i slike glavnog lika iz poznatog crtanog filma La linea radi se usporedba slikovnih formata u odnosu na kvalitetu i snagu kompresije. Usporedbom je ukazano da je JPEG prikladan za reljefne slike sa puno prijelaza i boja. Velika prednost ovog formata je što zauzima najmanje memorijskog prostora i gubitci prilikom kompresije nisu vidljivi ljudskim okom. Upravo iz tog razloga je popularan na webu jer se slike lako i brzo učitavaju na društvene mreže i blogove pa je fotografije lako dijeliti s drugim korisnicima. Ova osobina je karakteristična i za PNG. No, kako PNG format koristi kompresiju bez gubitaka ipak ga treba izbjegavati ukoliko se radi o fotografijama visoke rezolucije i kod ispisa grafike profesionalne kvalitete. U isto vrijeme GIF format se pokazao idealan za slike oštih linija i ograničene palete boja, te animacije. Navedeno je vidljivo na slikama glavnog lika iz spomenutog crtanog filma. TIFF format je najbolje primijeniti na ilustracije, profesionalne fotografije, prilikom skeniranja dokumenata visoke kvalitete, ispisa slika visoke kvalitete, za slike u časopisima i novinama. Glavna primjena BMP datoteke je za pohranjivanje visokokvalitetnih digitalnih fotografija, a kako koristi široki spektar boja i detalja idealan je za visokokvalitetne 2D digitalne fotografije.

11. POPIS LITERATURE

- [1] Awatj, R.: „DEFINITION lossless and lossy compression“, s Interneta, <https://www.techtarget.com/whatis/definition/lossless-and-lossy-compression>, 11. listopada 2021.
- [2] McAnlis, C., Haecky, A.: „Understanding Compression: Data Compression for Modern Developers“, O' Reilly Media, INC., Sebastopol, 2016.
- [3] Cvjetković, M., Slade – Šilović, M.: „Kompresija slike“, s Interneta, <http://spvp.zesoi.fer.hr/seminari/1999/ks/ks.htm>, 1999.
- [4] Mijić, J.: „Stupanj kompresije i utjecaj na kvalitetu WEBP formata slike“ Diplomski rad, Sveučilište u Zagrebu, Grafički Fakultet, Zagreb, 2014.
- [5] Oswal, S., Singh, A., Kumari, K.: „Deflate compression algorithm“, International Journal of Engineering Research and General Science Volume 4, Issue 1, January-February, 2016.
- [6] Barnwal, A., s Interneta, <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/>, 25. srpnja, 2022.
- [7] Galić, I., „Postupak kompresije mirne slike primjenom parcijalnih diferencijalnih jednadžbi“ Doktorska disertacija, Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički fakultet, Osijek, 2011.
- [8] Zeeh, C.: „The Lempel Ziv Algorithm“, s Interneta, <https://ece.uwaterloo.ca/~ece611/LempelZiv.pdf>, 16. siječnja, 2003.
- [9] Budhrani, D.: „How data compression works: exploring LZ77“ s Interneta, <https://towardsdatascience.com/how-data-compression-works-exploring-lz77-3a2c2e06c097>, 12. rujna, 2019.
- [10] Turlajić, E.: „Kompresija slike i videa“ s Interneta, <https://www.project-benefit.eu/eplatform/?courses=65&download=3%20lecture.pdf>, 06. ožujka, 2014.
- [11] „LZW – compression“, s Interneta, [Data Compression - LZW - compression \(stringology.org\)](https://www.stringology.org/), 22.08.2022.
- [12] Saikia, A., R.: LZW (Lempel–Ziv–Welch) Compression technique“, s Interneta, [LZW \(Lempel–Ziv–Welch\) Compression technique - GeeksforGeeks](https://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique-geeksforgeeks/), 08. studeni, 2021.
- [13] Juroš, P., Srbiš, M., Grubišić, I.: „Lempel – Ziv Vizualizacija“, Seminarski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [14] Jančić, M.: „Norme za kompresiju mirnih slika“, Diplomski rad br. 1857., Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 02. svibnja, 2002.

- [15] Cetin, A. E., Gerek, O.N.: „Vector Quantization“, Wiley Online library, 16. travnja, 2006.
- [16] Vrdoljak, A., Miletić, K., „Načela fraktalne geometrije i primjene u arhitekturi i građevinarstvu“, Stručni rad, Sveučilište u Mostaru, Građevinski fakultet, Mostar, 2019.
- [17] Al-Ani, M., Awad, F., H.: „The JPEG image compression algorithm“, International Journal of Advances in Engineering & Technology, ISSN: 22311963, Svibanj, 2013.
- [18] Dias, D.: „JPEG Compression Algorithm“, s Interneta, <https://medium.com/breaktheloop/jpeg-compression-algorithm-969af03773da>, 04. svibnja, 2017.
- [19] „RGB vs YcbCr“, s Interneta, [RGB vs YCbCr - ¿Cuál es mejor en una conexión HDMI? - Iván Andrei \(ivanandrei.com\)](https://ivanandrei.com), 03.08.2022.
- [20] Vukelić, B., „Novi algoritam za izradu percepcijskih sažetaka temeljen na izdvajanju atributa biometrijskih karakteristika“, Doktorski rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2018.
- [21] „PNG“, s Interneta, [PNG – Wikipedija \(wikipedia.org\)](https://wikipedia.org), 13.08.2022
- [22] Erdmann, C.: „Finally understanding PNG“, s Interneta, <https://compress-or-die.com/Understanding-PNG?fbclid=IwAR0h1vpigHo5CFf6w0FcW8c52vI9oTGQwUhsyGiRFGyWe3KZOjsnuAg3cjl>, 06. lipanj, 2021.
- [23] McAnlis, C.: „How PNG Works“, s Interneta, <https://medium.com/@duhroach/how-png-works-fl174e3cc7b7>, 06. travanj, 2016.
- [24] „GIF“, s Interneta: [GIF - Wikipedia](https://wikipedia.org), 14.08.2022.
- [25] Welch, T., A.: „A Technique for High Performance Data Compression“, IEEE Computer, Vol. 17, No. 6, 1984, pp. 8-19., lipanj, 1984.
- [26] Blackstock, S.: „LZW and GIF Explained“, s Interneta, <https://www.eecis.udel.edu/~amer/CISC651/lzw.and.gif.explained.html>, 1987.
- [27] Davies, K.: „TIFF compression? Use discretion!“, s Interneta, <https://openpreservation.org/blogs/compression-at-your-discretion/>, 09. veljača 2016.
- [28] Farena, N.: „Kompresija podataka“, Diplomski rad, Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku, Osijek, 2020.
- [29] Sanchez, J., Canton M.: „The PC Graphics Handbook“, CRC PRESS, Taylor & Francis e-Library, 2006.
- [30] Poddar, S.: „Image Compression“, s Interneta, [Swetank Poddar - Extended Essay.pdf](https://swetankpoddar.com), 17.08.2022.

- [31] „Priroda“, s Interneta, <https://images.app.goo.gl/dMqQ6EGBMHXFFFk7>, 25.08.2022.
- [32] „La linea“, s Interneta, <https://images.app.goo.gl/9ZhdwxrbQdoQ2Xe68>, 25.08.2022.
- [33] „Primošten“, s Interneta, <https://images.app.goo.gl/nUDcee4L7pmkdZre7>, 15.09.2022.
- [34] „Crno – bijela fotografija“, s Interneta, <https://3axis.co/download/eo0qjy91/>, 15.09.2022.

SAŽETAK

Posljednjeg desetljeća svjedočimo transformaciji u načinu komunikacije. Proces transformacije još uvijek traje i svakim danom napreduje, a uključuje sveprisutni Internet, mobilne komunikacije i video komunikaciju. Kompresija podataka mogućnost je tehnologije za svaki od navedenih aspekata multimedijske revolucije. Bez kompresije podataka ne bi bilo praktično objavljivati slike, video ili audio format na web. Također kompresija podataka je zaslužna i za pojavu digitalne televizije kao i za jednostavnu komunikaciju preko mobilnih uređaja. Prijenos i pohranjivanje velike količine podataka predstavljaju ozbiljan problem. U digitalnim sustavima ograničen je broj bitova po uzorku i brzina prijenosa, a s druge strane brzina prijenosa ograničava broj podataka koji se mogu prenijeti u jedinici vremena. S obzirom na navedeno, za prijenos i pohranjivanje slike potrebno je izvršiti kompresiju. Po definiciji kompresija je proces smanjivanja potrebnog fizičkog prostora za pohranu podataka kroz korištenje određenih metoda za pohranjivanje podataka. Postoje dvije vrste kompresije: kompresija s gubitcima i kompresija bez gubitaka. Razvojem tehnologije razvila se i digitalna fotografija (slika) pa tako i različiti načini njenog pohranjivanja. Kako bi se slika pohranila na računalo koriste se razne metode pretvorbe slike u binarne brojeve zato se može reći da postoje različiti formati za zapis slike. Najpoznatiji formati za pohranu slike su: JPEG, PNG, GIF, TIFF, BMP a najviše korišteni su JPEG i GIF.

Ključne riječi: Kompresija, formati, JPG, GIF, PNG, TIFF, BMP, komunikacija, digitalna fotografija, podatci, prijenos, pohranjivanje

SUMMARY

In the last decade, we have witnessed a transformation in the way of communication. The process of transformation is still ongoing and advancing every day, and includes the ubiquitous Internet, mobile communications and video communications. Data compression is a technology possibility for each of the aforementioned aspects of the multimedia revolution. Without data compression, it would not be practical to publish images, video or audio format to the web. Data compression is also responsible for the emergence of digital television as well as for simple communication via mobile devices. Transferring and storing large amounts of data is a serious problem. In digital systems, the number of bits per sample and the transmission speed are limited, and on the other hand, the transmission speed limits the amount of data that can be transmitted in a unit of time. Considering the above, it is necessary to perform compression to transfer and store the image. By definition, compression is the process of reducing the required physical space for data storage through the use of certain data storage methods. There are two types of compression: lossy compression and lossless compression. With the development of technology, digital photography (image) has also developed, as well as different ways of storing it. In order to store the image on the computer, various methods of converting the image into binary numbers are used, so it can be said that there are different formats for recording the image. The most famous image storage formats are: JPEG, PNG, GIF, TIFF, BMP, and the most used are JPEG and GIF.

Key words: Compression, formats, JPG, GIF, PNG, TIFF, BMP, communication, digital photography, data, transmission, storage

POPIS PRILOGA

<i>Slika 2.1 Primjer Run – lenght kodiranja [3]</i>	2
<i>Slika 2.2 Huffmanovo stablo[5]</i>	4
<i>Slika 2.3 Primjer Huffmanovog stabla – I korak [6]</i>	5
<i>Slika 2.4 Primjer Huffmanovog stabla – II korak [6]</i>	5
<i>Slika 2.5 Primjer Huffmanovog stabla - posljednji korak [6]</i>	6
<i>Slika 2.6 Princip LZ77 metode [10]</i>	7
<i>Slika 7.1 Primjer LZ77 algoritma [10]</i>	23
<i>Slika 9.1 Priroda, (BMP, 492 KB (491.521 bytes)), La linea (16 KB (14.188 bytes)) [31,32]</i>	27
<i>Slika 9.2 Priroda (PNG, 170 KB (169.709 bytes)), La linea (5 KB (5.488 bytes)) [31,32]</i> 27	
<i>Slika 9.3 Priroda (GIF, 209 KB (208.617 bytes), La linea (GIF, 9 KB (8.651 bytes)) [31,32]</i>	28
<i>Slika 9.4 Priroda (TIFF, 487 KB (487.404 bytes)), La linea (TIFF, 51 KB (50.884 bytes)) [31,32]</i>	28
<i>Slika 9.5 Priroda, (JPEG, 71 KB (71.087 bytes)), La linea (JPEG, 8 KB (7.833 bytes)) [31,32]</i>	29
<i>Slika 9.6 Primošten (BMP, 283 KB (280.470 bytes)), Crno - bijela fotografija (389 KB (387.156 bytes)) [31,32]</i>	29
<i>Slika 9.7 Primošten (PNG, 117 KB (116.839 byets)), Crno - bijela fotografija (78KB (76.169 bytes)) [31,32]</i>	30
<i>Slika 9.8 Primošten (GIF, 54KB (53.888 bytes)), Crno - bijela fotografija (GIF, 110KB (110.116 bytes)) [33,34]</i>	30
<i>Slika 9.9 Primošten (TIFF, 281 KB, 280.732 bytes), Crno - bijela fotografija (TIFF, 387KB (387.402 bytes)) [33,34]</i>	31
<i>Slika 9.10 Primošten (JPEG, 24KB (24.018 bytes)), Crno - bijela fotografija (JPEG, 1,2 MB (1.150.637 bytes)) [33,34]</i>	31
<i>Tablica 2.1 Huffmanova tablica [5]</i>	6
<i>Tablica 5.1 Metode filtriraja</i>	17
<i>Tablica 6.1 Inicijalizirana tablica nizova</i>	20
<i>Tablica 6.2 Tablica kodova</i>	21