SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 47

SAŽIMANJE SLIKA UPOTREBOM GENETSKIH ALGORITAMA

Vinko Bedek

Zagreb, lipanj 2010.

Sadržaj

Uvod

Genetski algoritmi su postali popularnim oruđem u današnjem svijetu rješavanja problema iz različitih domena. Iako postoje i druge metode optimiranja poznate javnosti, genetski algoritmi dobivaju najviše pozornosti, pojavljuju se na televiziji i u novinama. Financiranje projekata sa genetskim algoritmima nadmašuje financiranja projekata na temelju drugih metaheurističkih metoda [1]. Jedan od razloga je zasigurno i sam naziv, genetski. Privlačnost se ne može osporiti. Genetski algoritmi se često predstavljaju javnosti s argumentom kako će raditi zbog toga što preslikavaju mehanizme evolucije. Budući je evolucija s vremenom stvorila nas, prema nama vrhunac evolucije, takav argument se često prihvaća kao valjan bez obzira o kakvoj se kompleksnosti problema radi. Iako to naravno nije istina, genetski algoritmi su zaista moćno optimizacijsko oruđe kao što je pokazano u mnogim radovima ali i primjenama izvan akademske zajednice.

U ovom je radu istražena mogućnost primjene genetskih algoritama u području sažimanja. Traženjem radova o genetskim algoritmima u sažimanju dolazi se do velikog broja radova koji se bave fraktalnim sažimanjem. Upravo na tu tehniku se stavlja naglasak i u ovom radu. Fraktalno sažimanje u današnje doba predstavlja zastarjelu ideju, ideju koja zapravo u praksi nikada nije zaživjela zbog svojih mnogih problema vezanih uz primjenu izvan akademskih krugova. No, ideja upotrebe genetskih algoritama u sažimanju je vrlo privlačna pa je upravo zbog toga i odabrana za temu rada.

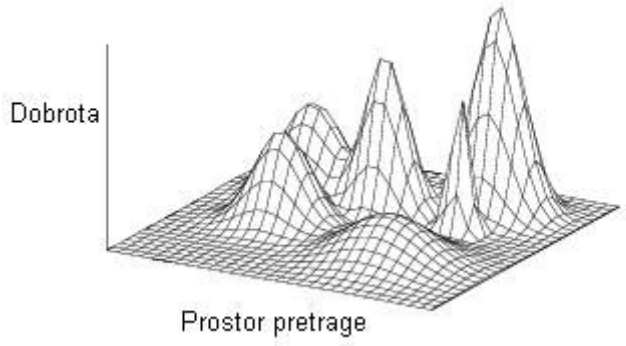
Rad se sastoji od 5 poglavlja. Nakon ovog, uvodnog, slijedi prvo poglavlje gdje se opisuju genetski algoritmi. Opisane su njihove najvažnije karakteristike, a dan je i kratak uvod u teoriju koja stoji iza njihovog ponašanja. U drugom poglavlju su opisane neke tehnike koje se koriste u sažimanju slika. Treće poglavlje daje matematičke osnove na kojima počiva fraktalno sažimanje. U četvrtom poglavlju je dan kratak opis upotrebe genetskih algoritama izvan fraktalnog sažimanja. Peto se poglavlje bavi programskim ostvarenjem fraktalnog kodera koji koristi genetske algoritme. Opisana su dva pristupa korištenja genetskih algoritama i dani su rezultati kodera u ovisnosti o parametrima kako genetskog algoritma tako i parametrima kodera nevezanih uz genetske algoritme. U dodatku se mogu pronaći primjeri slika koje su bile fraktalno sažete kao i njihovih originala.

1. Genetski algoritmi

Genetski algoritmi su općeniti algoritmi pretrage i optimizacije inspirirani evolucijom. Zbog mogućnosti primjene na široki skup problema, kao i zbog jednostavnih principa, genetski algoritmi dobivaju sve više sljedbenika iz različitih područja. Metode inspirirane evolucijom su postojale i prije genetskih algoritama. Ideja upotrebe populacije rješenja za rješavanje praktičnih inženjerskih problema je promatrana nekoliko puta kroz 1950 i 1960. Ingo Rechenberg je 1960ih stvorio evolucijske strategije za potrebe optimizacije funkcija realnih varijabli. Algoritam se sastojao od jednog roditelja i jednog djeteta, dijete je nastajalo mutacijom roditelja. Fogel, Owens i Walsh su stvorili evolucijsko programiranje koje je imalo rješenja kandidate prikazane uz pomoć deterministčkih konačnih automata. Nad njima se primjenjivala mutacija, odnosno algoritam je nasumično mijenjao dijagrame prijelaza stanja. No za postojanje genetskih algoritama u današnjem smislu je odgovoran John Holland. 1960ih je proučavao mehanizme prilagodbe u prirodnim sustavima i načine na koje bi mogao te mehanizme prenijeti na računalne sustave. Godine 1975. u knjizi *Adaptation in Natural and Artificial Systems* predstavlja genetske algoritme. Holland je slijedio terminologiju biologije, kodirana rješenja je nazvao kromosomima, osnovne jedinice kodiranih rješenja genima, a vrijednosti koje te jedinice mogu poprimiti alelima, mjesto gena u kromosomu lokus [2]. Za razliku od ostalih metoda inspiriranih evolucijom koje su naglašavale selekciju i mutaciju, Holland je uveo i stavio naglasak na križanje koje je s razvojem genetike i razumijevanjem strukture DNA sve više dobivalo na važnosti. Na početku genetski algoritmi nisu izazvali veće zanimanje. Doktorskim radom Kena De Jonga, studenta kojem je Holland bio mentor to se promijenilo. On je u radu stavio naglasak genetskih algoritama na optimizacijske probleme, a ne na sposobnost prilagodbe, nešto što je kod Hollandovog rada na adaptivnim sustavima imalo veću važnost. Danas se većina radova vezanih uz genetske algoritme zasniva na optimizaciji. Kasnije je De Jong, za kojeg možemo reći da je glavni "krivac" za takav pogled na domenu upotrebe genetskih algoritama više puta objašnjavao da oni nisu zapravo optimizatori funkcija [3].

1.1. Osnove genetskog algoritma

Prema Hollandu genetski algoritam se sastoji od populacije jedinki nad kojima se izvršava selekcija za određivanje jedinki koje su zaslužile sudjelovati u stvaranju nove generacije. Početna se populacija dobiva slučajno, te je disperzija početne populacije najveća kroz cijeli tok genetskog algoritma. Nova generacija se stvara križanjem roditelja koji su za to odabrani prema svojoj dobroti, odnosno sposobnosti preživljavanja u svojem staništu kod stvarne evolucije. U svrhu odabira mora postojati neki način, odnosno funkcija vrednovanja dobrote jedinke. Na temelju te funkcije možemo dobiti izgled prostora pretrage. Bolje jedinke će se nalaziti na brdima, a gore u dolinama. Primjer prostora pretrage za funkciju sa dvije varijable se nalazi na slici 1.1. Prostori pretrage nisu uvijek jednostavnog oblika, čak i za jednostavnije probleme mogu postojati mnogobrojni vrhovi jedan pored drugog odijeljeni sa dubokim dolinama.



Slika 1.1 Prostor pretrage

Cilj je naravno pronaći najviši vrh koji se naziva globalni optimum, dok se ostali vrhovi nazivaju lokalnim optimumima. Na novostvorenim jedinkama može doći do mutacije. Preživljavanjem boljih jedinki kod stvarne evolucije dolazi do prenošenja genetskog materijala zaslužnog za preživljavanje na novu generaciju, što za posljedicu ima poboljšanje prilagođenosti cijele generacije. Ista se stvar primjećuje i kod genetskih algoritama.

Osnovne građevne jedinice genetskog algoritma su:

- Populacija jedinki
- Selekcija
- Križanje
- Mutacija

Četvrti element prema Hollandu, inverzija, se danas rijetko koristi. Vidljivo je da su genetski algoritmi u svojoj osnovi jednostavni algoritmi čija prava snaga proizlazi kao i kod evolucije u paralelizmu i raznolikosti jedinki. Jedinke su zapravo kromosomi i predstavljaju jedno moguće rješenje.

Jednostavan genetski algoritam radi na sljedeći način:

1. Generira se početna populacija sa n slučajno odabranih jedinki iz prostora pretrage
2. Izračuna se dobrota svake jedinke
3. Ponavlja se dok se ne stvori n jedinki
    - Izabere se par jedinki iz trenutne populacije, vjerojatnost odabira ovisi o dobroti jedinke. Jedna jedinka može biti više puta odabrana. S određenom vjerojatnošću dolazi do križanja, novonastale jedinke ili jedinka se stavlja u novu generaciju. Ako do križanja nije došlo djeca su kopija roditelja.
    - Mutirati djecu s vjerojatnošću križanja
4. Zamijeniti trenutnu generaciju s novostvorenom
5. Provjeriti uvjet završetka, ako nije zadovoljen nastaviti sa 2. korakom

Svaka iteracija se naziva generacijom. Uvjet završetka može biti provjera konvergencije, neko vremensko ograničenje ili broj generacija.

1.2. Prikaz jedinki

Prije nego se može početi sa izradom genetskog algoritma potrebno je odlučiti kakav će se prikaz koristiti. U najjednostavnijem i najčešćem slučaju radi se o binarnom prikazu, koji se najranije koristio i uz koji je vezana većina teorije o

genetskim algoritmima. Iako najzastupljenije, ovo kodiranje ima slabosti zbog postojanja takozvanih Hammingovih litica gdje bliske jedinke unutar prostora pretrage imaju veliku Hammingovu udaljenost [4], npr. euklidska udaljenost između jedinke kodirane sa 011111 i jedinke sa prikazom 100000 je 1, ali Hammingova udaljenost je 6. Genetski algoritam treba kroz evoluciju invertirati sve bitove da bi došao do rješenja koje je minimalno bolje od postojećih. Za većinu problema koji se susreću u industrijsko-inženjerskom svijetu binarna reprezentacija nije dovoljna zbog otežane prilagodbe problema prikazu. Kod odabira prikaza važno je da zauzima što manje mjesta uz uvjet da sadrži sve važne informacije potrebne za dobivanje rješenja. Očito je da odabir prikaza ima ogroman utjecaj na daljnji razvoj algoritma. Na temelju kakve vrijednosti gen može poprimiti prikazi se mogu podijeliti na:

- Binarni prikaz
- Prikaz brojevima s pomičnim zarezom
- Prikaz cijelim brojevima
- Prikaz objektima, strukturama

Prikaz brojevima s pomičnim zarezom se pokazao superiornijim od binarnih kod funkcijskih optimizacija. Prikaz cijelim brojevima, odnosno permutacijski prikaz se najviše koristi kod kombinatoričkih optimizacijskih problema budući se traži najbolja permutacija. Prema strukturi prikaza, prikazi se dijele na jednodimenzionalne i višedimenzionalne, u većini slučajeva se koristi jednodimenzionalni prikaz, ali za neke probleme je prirodnije koristiti višedimenzionalne. Genetski algoritmi rade sa dva tipa prostora. Jedan je prostor kodiranja, ili genotipski prostor, a drugi je prostor rješenja, ili fenotipski prostor. Mutacija i križanje rade nad genetskim prostorom, a selekcija i evaluacija jedinki rade nad prostorom rješenja. Između tih prostora postoji preslikavanje koje može biti 1:N, N:1 i 1:1. Kod preslikavanja se može ustvrditi da se neki kromosom iz genotipskog prostora ne može preslikati u fenotipski. Takav kromosom je ilegalan. Ilegalnost potječe iz prikaza i upotrebe genetskih operatora ako koristimo prikaz prilagođen problemu i operatore koji to nisu. U tom slučaju možemo ili odbaciti ilegalne jedinke, ili popraviti jedinke na neki način. Drugo rješenje je koristiti genetske operatore prilagođene problemu.

Svojstva na temelju kojih se određuje kvaliteta prikaza [4]:

- Neredundantnost

  Preslikavanje između genotipskog i fenotipskog prostora bi trebalo biti 1:1. Ako imamo N:1 preslikavanje trošimo vrijeme na traženje među istovjetnim rješenjima, jedinke su različite u genotipskom prostoru pa genetski algoritam ne zna da su te jedinke jednake. Zbog takve situacije u algoritmu može doći do prerane konvergencije. Najgori je slučaj kada postoji 1:N preslikavanje jer treba odrediti koje će se rješenje iz fenotipskog prostora uzeti.

- Legalnost

  Bilo koja permutacija prikaza odgovara nekom mogućem rješenju. Ovim svojstvom osigurano je korištenje postojećih operatora. Jasno je da se ovo svojstvo na razini prikaza za složenije probleme teško zadovoljava.

- Kompletnost

  Ovo svojstvo govori da se do bilo koje točke prostora rješenja može doći pretragom genetskim algoritmom. Kada prikaz ne bi posjedovao takvo svojstvo postojala bi mogućnost da se kvalitetna rješenja nalaze upravo u nepristupačnom dijelu prostora pretrage.

- Lamarckianovo svojstvo

  Značenje alela za neki gen je neovisno o kontekstu. Tim svojstvom je osigurano da prinos kvaliteti rješenja zbog tog gena neće opasti mijenjanjem ostalih gena kroz proces evolucije.

- Kauzalnost

  Male varijacije nastale u genotipskom prostoru uslijed mutacije bi trebale rezultirati malim varijacijama u fenotipskom prostoru, drugim riječima mutacija bi trebala očuvati strukturu susjedstva (jedinki sličnih promatranoj po nekom kriteriju) u fenotipskom prostoru. Procesi pretrage koji ne uništavaju susjedstva posjeduju jaku kauzalnost, dok slaba kauzalnost znači veće promjene u fenotipskom susjedstvu jedinke malim promjenama u genotipskog prostora.

1.3. Teorem shema

Kod nekih ostalih heurističkih pristupa, npr. simuliranog kaljenja, postoji znatna količina teoretske analize. Osim što je važna za shvaćanje rada, teorija može dati smjernice u implementaciji i području uporabe [1]. Broj primjena genetskih algoritama raste sa njihovom popularnošću, no razvoj teorije genetskih algoritama je puno sporiji. Teorija o načinu rada genetskog algoritma je još uvijek fragmentirana i postoje različite perspektive. Najstarije teoretsko razmatranje koja se odnosi na originalni način rada genetskog algoritma dao je Holland. U svom radu Holland uvodi pojam shema. Sheme se odnose na bitovne nizove, a sastoje se od 0 , 1 i *. Npr. shema ***0**1 predstavlja sve jedinke koje imaju četvrti gen 0 i zadnji gen 1. Jedinke koje zadovoljavaju shemu se nazivaju instance sheme. Red sheme je broj nepromjenjivih bitova, a u slučaju prethodnog primjera iznosi 2. Najveća udaljenost između nepromjenjivih bitova unutar sheme se naziva definirajući razmak. Razlog uvođenja shema je bio u svrhu formalnog opisivanja ideje građevnih jedinica pomoću kojih genetski algoritam dolazi do rješenja. Na temelju shema i proučavanja rada genetskog algoritma nastao je teorem sheme. Teorem sheme kaže da broj jedinki koje sadrže shemu niskog reda, kratkog definirajućeg razmaka te iznadprosječne dobrote raste eksponencijalno [5]. Spomenuta dobrota sheme se računa implicitno kod računanja dobrote instanci te sheme unutar populacije. U teoremu shema su sadržana destruktivna svojstva mutacije i križanja. Zašto sheme niskog reda i kratkog razmaka? Ako imamo roditelje sa istom shemom, križanjem ćemo dobiti opet istu shemu, no ako su sheme različite postoji vjerojatnost da novonastala jedinka neće biti instanca sheme niti jednog roditelja. Što je definirajući razmak manji veća je vjerojatnost da će shema biti očuvana i da će se svi bitovi koji određuju shemu križanjem biti preneseni na dijete. Razlog za rast shema sa niskim redom je u primjeni mutacije. S većim brojem bitova koji moraju ostati nepromijenjeni kako bi shema ostala očuvana raste vjerojatnost da će jedan od njih biti promijenjen mutacijom. Na temelju teorema shema dolazi se do zaključka da genetski algoritam dolazi do rješenja sastavljajući kratke sheme, odnosno da kratke sheme s iznadprosječnom dobrotom predstavljaju građevne blokove. Navedeni zaključak se naziva hipoteza građevnih blokova. Točna definicija glasi: Genetski algoritam pretražuje prostor rješenja nizajući sheme niskog reda, kratke definirane dužine i iznadprosječne

dobrote, zvane građevni blokovi [5]. Teorem shema je od svog nastanka višestruko kritiziran. Michael Vose je proučavajući utjecaj mutacije kod genetskog algoritma pokazao da malena promjena u vjerojatnosti primjene mutacije može imati značajan utjecaj na daljnji put genetskog algoritma kroz prostor pretrage koji analiza preko shema ne može objasniti. Vose isto tako navodi da je fokus na broj instanci jedinki neke sheme neispravan. Za genetski algoritam je puno važnije koje instance sheme se pojavljuju u sljedećoj generaciji [1]. Pod provjerom se našla i hipoteza građevnih blokova. Goldberg je predstavio ideju *deceptivnih* funkcija, funkcija koje bi navele genetski algoritam na krivi put uskraćujući dobre građevne blokove. Očekivano, genetski algoritam radi loše, i daje dokaz o ispravnosti hipoteze. S druge strane ako se promatra rad genetskog algoritma nad "*Royal Road*" funkcijom može se doći do drugačijih zaključaka. "*Royal Road*" funkcija daje genetskom algoritmu sve potrebno da dođe do rješenja na način koji je očekivan prema hipotezi. Rezultati su bili porazni za genetski algoritam, na problemu gdje je trebao pokazati snagu građevnih blokova je po pitanju performansi poražen od najjednostavnije determinističke metode lokalne pretrage (engl. *hill-climber)*[1].

1.4. Selekcija

Kao i u prirodi uloga selekcije kod genetskog algoritma je odabrati jedinke koje će sudjelovati u stvaranju nove populacije. Cilj je stvoriti populaciju koja je bolja od stare, odnosno sastoji se od jedinki sa boljom dobrotom. Kako bi se to moglo postići, genetski materijal dostupan za stvaranje te generacije mora biti kvalitetan. Ta osobina se ostvaruje davanjem veće vjerojatnosti jedinkama sa većom dobrotom za sudjelovanje u stvaranju nove generacije. Treba naglasiti da se u svrhu dobivanja što bolje jedinke ne smiju zanemariti one sa lošijom dobrotom, jer postoji mogućnost da se u njima nalazi genetski materijal koji može pogodovati boljoj dobroti kod djeteta, slično kao i u prirodi. Isto kao što postoji mogućnost gubitka dobrog genetskog materijala koji se nalazi u lošim jedinkama zbog smanjene vjerojatnosti sudjelovanja u reprodukciji novih jedinki, postoji i mogućnost da izgubimo najbolju jedinku budući sve stare jedinke nestaju. Iako je genetski materijal prenesen, ne postoji garancija da u novoj generaciji imamo jedinku koja je po svojoj dobroti jednaka najboljoj u prijašnjoj generaciji. Dolazi se

do ideje da prenesemo najbolju, ili nekoliko najboljih jedinki u novu generaciju. Taj mehanizam se naziva elitizam.

Selekcije dijelimo na generacijske i eliminacijske. Kod generacijskih selekcija odabiremo bolje jedinke koje će sudjelovati u reprodukciji. Za vrijeme stvaranja nove generacije potrebno je zadržati staru za izbor roditelja. U eliminacijskoj selekciji brišemo određeni broj jedinki. Odabir jedinki koje se brišu je težinski. Hollandov genetski algoritam je koristio jednostavnu proporcionalnu selekciju. Vjerojatnost da će jedinka biti odabrana je proporcionalna njezinoj dobroti. Vizualno se metoda može predočiti kao kotač ruleta gdje svaka jedinka ima svoj udio na temelju dobrote. Implementacija je sljedeća:

- Sumirati dobrote svih jedinki, T
- Ponavljati za broj željenih jedinki
    - Izabrati nasumičan broj između 0 i T, R
    - U petlji prolaziti kroz sve jedinke i sumirati dobrote, kada suma postane veća od R , zadnja jedinka čiju smo dobrotu zbrojili je odabrana

U svrhu ubrzanja se kod jednostavne selekcije umjesto opisanog, linearnog načina pretrage može koristiti binarno pretraživanje. Statistički, kod ove metode svaka jedinka u generaciji ima broj djece određen dobrotom. No, veličina populacije kod genetskih algoritama često nije dovoljno velika kako bi se statistički model preslikao u stvarnu situaciju. Godine 1987. James Baker je predložio stohastičku univerzalnu metodu. Odabir roditelja se vrši samo jednom na temelju slučajnog broja, a *n* roditelja se odabire deterministički. Najveći problem kod proporcionalnih selekcija leži u tome što na početku genetskog algoritma postoji malen broj jedinki koje dobrotom odudaraju od prosjeka. Primjenom proporcionalne selekcije algoritam stavlja preveliki naglasak na pretraživanje prostora pretrage u kojem su te jedinke. Isto tako, kasnije kada jedinke postaju sve sličnije i razlike u dobroti su male, algoritam ne stavlja dovoljan naglasak na odabir boljih jedinki. Može se reći da brzina evolucije ovisi o varijanci dobrote u populaciji. U tu svrhu uvode se skalirajuće metode. Od vremena De Jongovih radova primjena skalirajućih metoda je postala široko prihvaćena [4]. Predloženo je nekoliko načina skaliranja koji se dijele na statičke i dinamičke skalirajuće metode. Ako je relacija transformacije

konstantna tada se radi o statičkim metodama, ako je pak relacija podložna promjeni u ovisnosti o nekim vremenski promjenjivim faktorima govorimo o dinamičkim skalirajućim metodama. Primjer jedne dinamičke skalirajuće metode je sigma skaliranje. Jedna mogućnost ostvarivanja je prikazana preko jednadžbe (1.1) [2].

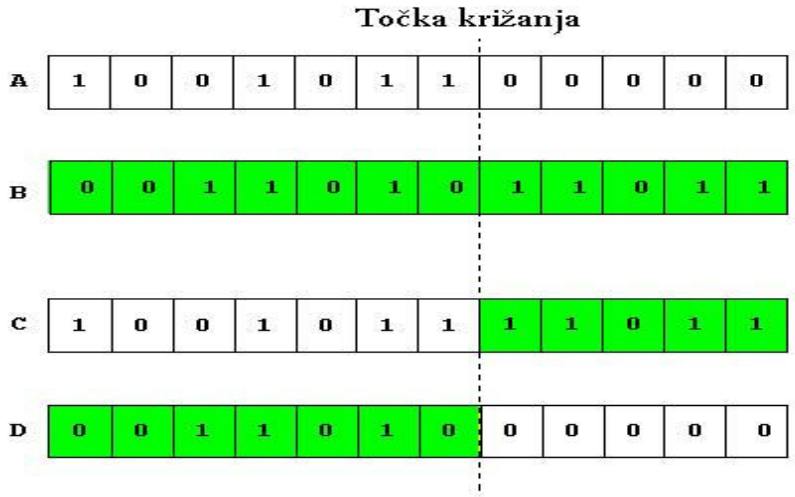
$$\mathrm{ExpVal}(i,t) = \begin{cases} 1 + \dfrac{f(i) - \bar{f}(t)}{2\sigma(t)} & \text{if } \sigma(t) \neq 0 \\ 1.0 & \text{if } \sigma(t) = 0, \end{cases} \qquad (1.1)$$

gdje *ExpVal(i, t)* označava skaliranu vrijednost dobrote jedinke *i* u promatranom vremenu *t*, *f(i)* trenutnu dobrotu jedinke s indeksom *i,f̄(t)* srednju vrijednost dobrote unutar generacije, te σ*(t)* standardnu devijaciju dobrote za trenutnu generaciju.

Iz formule je vidljivo da će selekcijski pritisak kroz tok genetskog algoritma ostati ujednačen. Budući se promatra razlika srednje dobrote i dobrote jedinke skalirana sa standardnom devijacijom na početku algoritma bolje jedinke neće biti jako udaljene od srednje vrijednosti, odnosno kod selekcije će vjerojatnost njihovog odabira biti manja. Kasnije, kada standardna devijacija opadne, bolje jedinke će imati veću vjerojatnost odabira od lošijih nego što bi to imale kod selekcije samo na temelju dobrote. Sljedeći način kojim se izbjegava prerana konvergencija je rangirajuća selekcija. Vjerojatnost odabira ovisi o rangu jedinke unutar populacije. Jasno je da za posljedicu algoritam može sporije doći do rješenja, ali isto tako zadržavanje raznolikosti unutar generacije može dovesti do kvalitetnijeg rješenja. Još jedan nedostatak rangirajućih selekcija je vrijeme potrebno za sortiranje cijele populacije. Metoda koja je efikasnija od rangirajućih, a zadržava dobra svojstva vezana uz selekcijski pritisak (udio djece u generaciji koji će potjecati od roditelja sa visokom dobrotom) se naziva turnirska. Odabire se k jedinki te se najbolja uzima za roditelja. Kod eliminacijske turnirske selekcije se odabiru 3 ili više jedinki, a jedinka s najmanjom dobrotom se zamjenjuje sa djetetom dvije najbolje jedinke.

1.5. Križanje

Križanjem se stvaraju nove jedinke iz genetskog materijala roditeljskih kromosoma. Jednim križanjem se mogu stvoriti dvije nove jedinke ili samo jedna. Samo križanje je izvedeno na način da zadovoljava prikaz kromosoma. Najjednostavnije križanje je križanje s jednom točkom prekida prikazano na slici 1.2. Kromosom jednog roditelja se prekida u jednoj točki te se na prekinuti dio nadovezuje preostali dio iz drugog roditelja.



Slika 1.2 Križanje sa jednom točkom prekida

Ova vrsta križanja ima svojih nedostataka. Ne postoji mogućnost kombiniranja svih shema, npr. ako imamo shemu jednog roditelja 1****1 i drugog ***1** nikako križanjem ne možemo dobiti shemu 1**1*1. Isto tako sheme sa relativno dugim definirajućim razmakom u odnosu na ukupnu duljinu kromosoma će s vremenom nestati jer će ih operator križanja uništiti, jer prema teoremu shema algoritam sa jednostavnim križanjem daje prednost shemama sa manjim definirajućim razmakom. Naziv pojave je pozicijska sklonost i točna definicija glasi: sheme koje se mogu stvoriti ili uništiti križanjem ovise o poziciji bitova unutar kromosoma [2]. Davanje prednosti shemama sa manjim definirajućim razmakom ima još jednu posljedicu. Bitovi koji su blizu bitova koji čine željenu shemu imaju veću vjerojatnost za očuvanje kroz tijek algoritma. Osim toga primijećeno je da ovakvo križanje zadržava bitove na krajnjim pozicijama budući da je najmanji dio kromosoma koji se može izmjenjivati prvi ili zadnji bit. U svrhu eliminiranja ovih pojava Holland je u svom originalnom genetskom algoritmu koristio operator

inverzije. Svrha inverzije nije bila stvaranje novog genetskog materijala, već samo preslagivanje gena na drugačiji način za izbjegavanje gore spomenutih efekata križanja sa jednom točkom prekida. Jedinka prije i poslije inverzije ne izgledaju isto sa stanovišta križanja i ako ih križamo sa nekom drugom jedinkom rezultati neće biti isti. Nakon operatora inverzije jedinka zadržava sva svoja svojstva. Način na koji se to postiže je pridjeljivanje svakom genu broj koji označava lokus na kojem se nalazi.

Zbog svojih nedostataka u očuvanju shema kao i ograničenju u stvaranju novih koristi se križanje s dvije točke prekida gdje se točke odabiru nasumično a segmenti unutar tih točaka se kod roditelja zamjenjuju. Osim križanja sa dvije točke prekida moguće je koristiti bilo koju vrijednost između 1 i n-1 gdje je n broj gena. Ako se koristi križanje sa n-1-om točkom prekida tada se radi o uniformnom križanju s maskom 1010101..., naizmjence se uzimaju geni prvo iz jednog roditelja pa iz drugog. Križanje koje koristi masku sa jednakim vjerojatnostima se naziva p-uniformno križanje gdje je p vjerojatnost da se gen uzme iz prvog roditelja. Za općenitiju slučaj sa p=0.5 se koristi samo naziv uniformno križanje. Opisane vrste križanja se odnose na bitovne prikaze, i iako se mogu iskoristiti i za druge prikaze u većini slučajeva je potrebno pronaći ili stvoriti križanje koje je prilagođeno vrsti problema koji se pokušava riješiti.

1.6. Mutacija

Primjenom samo križanja nad jedinkama se kroz određeni broj generacija dolazi do prerane konvergencije, genetski materijal koji je algoritmu na raspolaganju postaje homogen, te nema raznolikosti između jedinki. Kao što vrste u prirodi ne bi preživjele bez mutacija nad genima, odnosno prilagodbi okolini, tako niti jedinke u genetskom algoritmu neće dosegnuti svoj potencijal jer će zaglaviti na nekom lokalnom optimumu unutar prostora pretrage. Bitovi na pojedinim mjestima unutar svih jedinki će postati isti i, ako rješenja sa boljom dobrotom na tim mjestima imaju drugačiji raspored nula i jedinica, ne postoji način da algoritam dođe do tih rješenja bez upotrebe mutacije. Iako je uloga križanja neupitna vidljivo je da bez mutacije efikasan genetski algoritam ne može postojati. Najjednostavnija mutacija kod binarnog prikaza je invertiranje bita. Evaluacija o izvršavanju operatora mutacije se kod takvog prikaza obavlja za svaki bit i radi se o vjerojatnostima od 0.005 do

0.01. Miješajuća mutacija za razliku od jednostavne mutacije radi nad kromosomima, a ne nad genima; odabiru se dvije pozicije unutar kromosoma i svi geni između se ispremiješaju, ako se geni nanovo generiraju radi se o potpunoj miješajućoj mutaciji. Potpuna mutacija izmiješa sve bitove nekog slučajno odabranog kromosoma. Ako se radi o permutacijskom prikazu za potrebe nekog kombinatoričkog problema, najjednostavnija mutacija koja se može primijeniti je zamjena pozicija dvaju gena. Napredniji primjer mutacije kod permutacijskog prikaza bi bio PBM (*Position Based Mutation*). Uzima se gen s neke lokacije i stavlja se na neko drugo nasumično odabrano mjesto.

Kromosom prije PBM: 1 6 4 9 3 **7** 2 5 8 0

Kromosom poslije PBM: 1 **7** 6 4 9 3 2 5 8 0

1.7. Parametri

Za zadovoljavajuće rezultate i zadovoljavajuću brzinu pronalaska takvih rezultata potrebno je pronaći odgovarajuće vrijednosti parametara. Osnovni parametri koji se podešavaju su veličina populacije, vjerojatnost križanja i vjerojatnost mutacije. Bez previše razmišljanja se dolazi do nekih osnovnih zaključaka o veličinama određenih parametara. Premalen broj jedinki nije dovoljan za održavanje potrebne količine genetskog materijala, prevelik broj s druge strane bespotrebno usporava izvođenje. Vjerojatnost mutacije mora biti malih razmjera ako želimo zadržati konvergenciju, preveliki iznos tjera algoritam na nasumično kretanje prostorom pretrage, maleni iznosi vjerojatnosti križanja sprječavaju kombiniranje dobrih svojstava roditelja. Izvršeno je mnogo istraživanja o optimalnim vrijednostima parametara, no kako je svaki algoritam drugačiji i vrijednosti koje će davati najbolja rješenja će se razlikovati od slučaja do slučaja. Jedan od ranijih istraživanja o vrijednostima parametara je izvršio De Jong, a prema njegovim rezultatima najbolja veličina populacije je između 50 i 100 jedinki, vjerojatnost križanja oko 0.6, a vjerojatnost mutacije je odredio kao 0.001 po bitu. Ovi parametri su prihvaćeni u javnosti kao polazni za različite vrste problema. 1986 Grefenstette je uz pomoć genetskog algoritma odlučio pronaći optimalne vrijednosti parametara. Radio je nad De Jongovim setovima problema, te je kao parametre koristio De Jongove preporuke. Kao rezultat je dobio jedinku koja je postavila veličinu populacije na 30

jedinki, vjerojatnost križanja na 0.95, vjerojatnost mutacije na 0.01, generacijski jaz (udio populacije koji se mijenja novim jedinkama) je bio 1, a koristila je i elitizam [2]. Osim fiksnih iznosa moguće je napraviti i adaptivni genetski algoritam koji će mijenjati vrijednosti parametara u skladu sa situacijom u kojoj se pronašao. U slučaju da postoji prevelika raznolikost unutar populacije algoritam može smanjiti vjerojatnost mutacije i povećati vjerojatnost križanja, a ako se nađe na području lokalnog optimuma, povećanjem vjerojatnosti mutacije će se preseliti na neki drugi podskup mogućih rješenja.

2. Sažimanje slika

Koliko god napredovali diskovi i brzina Internet veze, potreba za sažimanjem će postojati. Cilj će uvijek biti što učinkovitije pohraniti i prenijeti podatke, u ovom slučaju slike. Jedna slika veličine 1680x1050 u boji će nesažeta zauzimati oko 5 megabajta, što je za korisnika koji pretražuje internet u potrazi za pozadinu radne površine previše, čak i s 4Mbit vezom.

Nije teško primijetiti da i unutar najjednostavnije digitalne reprezentacije slika postoji dosta redundancije kao i informacija koje možemo smatrati nevažnima. Redundancija može biti prostorna ili spektralna, dok kod videa možemo primijetiti i temporalnu redundanciju. Neke informacije unutar slike se pak mogu izostaviti jer ih krajnji korisnik, ljudsko biće, ne može primijetiti zbog ograničenja u mogućnostima ljudskog vizualnog sustava.

Tehnike sažimanja se mogu podijeliti na sažimanje s gubitkom i sažimanje bez gubitka. Iz imena je očito da je razlika u tome hoće li reproduciran podatak biti jednak originalu ili neće. No, sažimanjem bez gubitka se ne mogu postići dovoljno visoki stupnjevi sažimanja. Neke od tehnika koje se koriste unutar algoritama sažimanja slika se nalaze u nastavku.

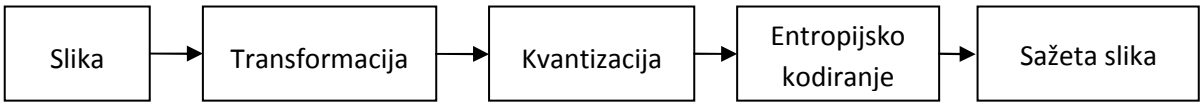
2.1. Diferencijalna pulsno kodna modulacija

Pulsno-kodna modulacija (engl. *Pulse-Code Modulation,* PCM) predstavlja uzorkovanje analognog signala i kvantizaciju dobivenih vrijednosti. Sažimanje je rezultat kvantizacije uzoraka. Diferencijalna pulsno-kodna modulacija (engl. *Differential Pulse-Code Modulation,* DPCM) postiže sažimanje iskorištavanjem lokalne korelacije između uzoraka. Tehnika se sastoji u tome da se koristi predikcija sljedećeg piksela, a razlika između stvarne vrijednosti i predviđene vrijednosti se kvantizira i kodira. Za visoko korelirane slike dobro dizajnirano predviđanje će rezultirati vrijednostima grešaka u malom opsegu, varijanca će biti manja. Druga opcija je uzimanje razlike trenutnog i prethodnog uzorka. Ako

smanjimo broj bitova po pikselu, greške uvedene zbog kvantizacije će biti manje nego kod PCM-a.

2.2. Kodiranje transformacijama

Slika se transformira linearnim operacijama čime se podaci slike pretvaraju u skup koeficijenata. Inverznim transformacijama možemo dobiti originalnu sliku. Kodiranje transformacijama se ne obavlja nad cijelom slikom, već nad blokovima fiksne veličine. U području sažimanja slika su se najzanimljivijima pokazale ortonormalne transformacije kao Karhunen-Loeve, diskretna kosinusna, Walsh-Hadamard, te diskretna Fourierova transformacija. Istraživanja nad prirodnim slikama su pokazala da je većina energije slike koncentrirana u području niskih frekvencija. Prema tome, koeficijenti koji pripadaju visokim frekvencijama se mogu ili odbaciti ili kvantizirati. Najbolja transformacija u smislu kompaktnosti energije se pokazala Karhunen-Loeve, no zbog činjenice da ne postoji brzi algoritam, u području sažimanja slika i videa se najviše koristi DCT **Error! Reference source not found.**. Kodiranje transformacijama samo po sebi nije sažimanje, već samo prebacivanje u drugu domenu. Nakon transformacije potrebno je provesti dodatne korake kako bi stvarno saželi sliku. Provodi se kvantizacija i entropijsko kodiranje. Redoslijed operacija je vidljiv na slici 2.1.



Slika 2.1 Koraci kod upotrebe kodiranja transformacijama

2.3. Kodiranje valićima

Valići su u svojoj osnovi funkcije definirane na konačnom intervalu sa srednjom vrijednošću 0. Ideja valić transformacije je predstaviti bilo koju funkciju kao zbroj skupa valića. Valići unutar tog skupa se pak dobivaju primjenom operacija skaliranja, translacije i širenja/skupljanja osnovnog valića koji se naziva *mother wavelet.*

Iako korištenje valića spada pod transformacijsko kodiranje, zbog svoje uspješnosti u novije doba svrstano je u zaseban dio. Upotreba im nije ograničena na područje sažimanja slika, služe i za uklanjanje šuma u audio zapisima, sažimanje signala, prepoznavanje objekata, dijagnosticiranje srčanih bolesti, prepoznavanju govora. Osnovna ideja kodiranja valićima je podjela slike na aproksimacijski dio i na dio s detaljima (slika 2.2 a) . Dio s detaljima se sastoji od horizontalnog dijela, vertikalnog i dijagonalnog. Dekompozicija se za aproksimacijski dio može rekurzivno nastaviti (slika 2.2 b). Nakon dekompozicije može se primijeniti kvantizacija, ili zanemarivanje neželjenih nivoa detalja [7] .

(a) (b)

Slika 2.2 Valić dekompozicija a)jedna razina, b) dvije razine

2.4. Vektorska kvantizacija

Ideja vektorske kvantizacije leži u Shannonovom otkriću da se efikasnije kodiranje može postići kvantizirajući promatrane uzorke kao vektore a ne kao skalare. Slika se dijeli na blokove određene veličine te se zatim svaki blok uspoređuje s skupom reprezentativnih vektora. Sažimanje se sastoji u pamćenju indeksa najsličnijeg reprezentativnog vektora. Performanse vektorske kvantizacije ovise o skupu reprezentativnih vektora. Ključan element vektorske kvantizacije je dizajniranje što boljeg rječnika (engl. *Codebook*). Rječnik se generira na temelju uzoraka za treniranje. Posebnim algoritmom se odabere početni rječnik, a zatim se nekim od algoritama grupiranja(engl. c*lusteringa*), primjerice pomoću generaliziranog

Lloydovog algoritma, dolazi do konačnog rječnika. Za najbolje rezultate potreban je veliki rječnik [8].

Brzine izvođenja kodera i dekodera su asimetrične, kod kodiranja je potrebno usporediti trenutni blok s vektorima iz rječnika, dok je kod dekodera jedina potrebna operacija čitanje indeksa i kopiranje vektora iz rječnika s tim indeksom na pravilno mjesto.

2.5. Entropijsko kodiranje

Entropija predstavlja donju granicu prosječnog broja bitova potrebnih za reprezentiranje simbola. Drugim riječima, ona predstavlja granicu sažimanja bez gubitka. Entropijsko kodiranje se zasniva na neuniformnoj raspodjeli pojave određenih simbola unutar podataka. U popularne metode spadaju Huffmanovo kodiranje, aritmetičko kodiranje, te kodiranje duljine niza (engl. *run lenght encoding*). Kodiranje duljine niza se često koristi u kombinaciji s Huffmanovim kodiranjem, kao što je to slučaj u JPEG-u. Predstavlja jednostavnu tehniku kodiranja i temelji se na ponavljanju simbola. Niz simbola se mijenja s zapisom o kojem se simbolu radi i koliko se puta ponovio. Naravno, postupak je efikasan kad ima dovoljno ponavljanja. Jedna od primjena je u faksevima. Isto tako upotrebljava se kod JPEG-a, gdje se koristi modificirana verzija koja broji samo nule, odnosno koeficijente nakon transformacije i kvantizacije kojima je vrijednost 0. Huffmanovo kodiranje se temelji na zamjeni simbola fiksne duljine s kodnim riječima varijabilne duljine. Duljina kodne riječi ovisi o vjerojatnosti kojom se pripadni simbol pojavljuje u izvoru (nesažetim podacima). Simboli koji se češće pojavljuju dobivaju kraće kodne riječi. Kod aritmetičkog kodiranja se ne koriste kodne riječi za specifične simbole, već je cijeli ulaz reprezentiran jednom kodnom riječi koja specificira broj između 0 i 1. Svaki simbol ulaza je reprezentiran jednim intervalom. Dolaskom simbola iz izvora, odabire se pripadni podinterval, a svaki sljedeći simbol rezultira odabirom užeg podintervala koji jedinstveno identificira dotadašnji niz simbola. Aritmetičko kodiranje daje bolje rezultate sažimanja, ali Huffmanovo kodiranje je zbog svoje jednostavnosti i visoke brzine šire zastupljeno. Možemo ga primjerice pronaći u JPEG-u, MP3-u, te *deflate* algoritmu sažimanja.

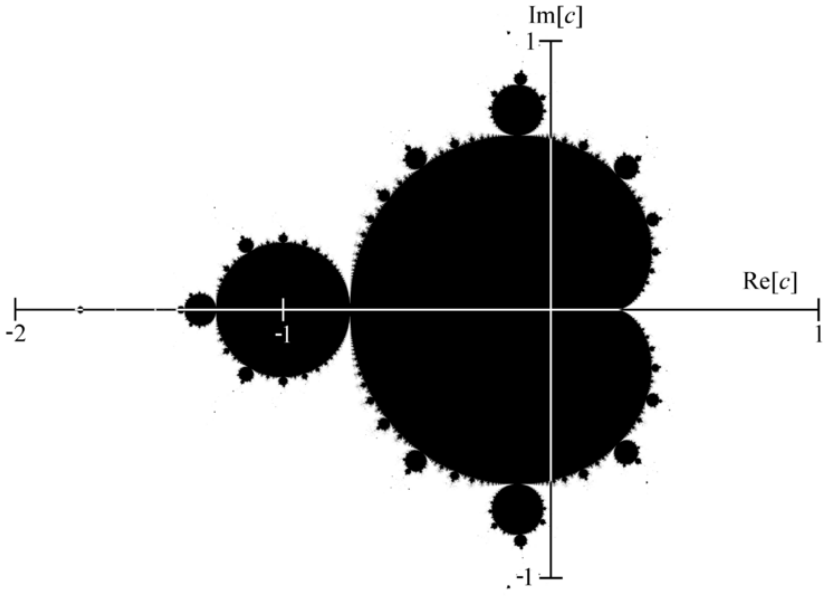
3. Fraktalno sažimanje

Fraktal je termin koji je uveo Mandelbrot koji specificira geometrijski lik gdje je svaki dio tog lika usporediv s umanjenom kopijom cjeline. Fraktalni objekti su samoslični i kao posljedica te samosličnosti neovisni o skaliranju.

Mandelbrotov skup je skup točaka u kompleksnoj ravnini čija granica formira fraktal. Sama formula je jednostavna i opisana je izrazom 3.1.

$$z_{n+1} = {z_n}^2 + c \tag{3.1}$$

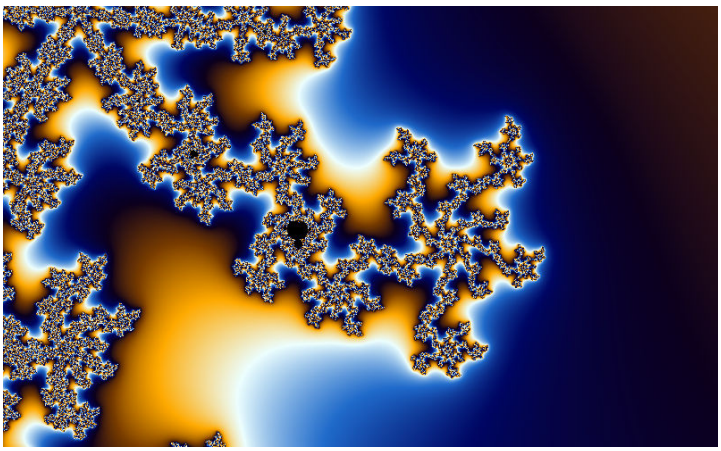
Za kompleksni broj *c* se kaže da je unutar Mandelbrot skupa ako bez obzira na broj iteracija *n* apsolutna vrijednost od *z(n)* ne prijeđe neku granicu, uzimajući z(0)=0. Uvrštavajući *c*=1 vidimo da 1 ne spada u Mandelbrot set jer niz teži u beskonačno. Računajući za kompleksne brojeve i crtanjem u kompleksnoj ravnini može se primijetiti kompleksna granica koja se ne pojednostavljuje s povećanjem.



Slika 3.1 Mandelbrotov skup

Na slici 3.1 je crnom bojom prikazan skup Mandelbrot serije. Osim ovog jednostavnog da/ne slučaja mogu se dobiti i kompleksnije slike pridruživanjem boje kompleksnom broju u ovisnosti o iteraciji u kojoj se utvrdilo da ne pripada

Mandelbrot seriji i samom broju. Primjer se može vidjeti na slici 3.2 gdje je prikazan detalj Mandelbrotovog fraktala.



Slika 3.2 Detalj Mandelbrotovog fraktala, boja predstavlja brzinu divergencije

Mandelbrot nije uzeo u obzir mogućnost upotrebe fraktala u području sažimanja, ali je pokazao da mogu biti upotrijebljeni za približan opis mnogih objekata u svijetu kao obala, planina, drveća, oblaka.

Primjenu fraktala na područje sažimanja slika je kao potencijal prepoznao Barnsley [9]. Nakon objave knjige *Fractals Everywhere*, fraktalno sažimanje je dobilo znatnu popularnost. Jedna od slika je poznata Barnslayeva paprat (slika 3.3) koju je prvi puta opisao u spomenutoj knjizi. Za generiranje slike koristi se iterativni funkcijski sustav (IFS) od 4 linearne transformacije, te postoji posebni algoritam konstrukcije. Fraktalno sažimanje je dobilo na popularnosti djelomično i zbog tvrdnji Barnsleya o omjerima sažimanja od 10000:1, no takvi visoki omjeri sažimanja su bili mogući samo za specijalne slučajeve, kao i uz potrebnu ljudsku intervenciju.

Slika 3.3 Barnsleyeva paprat

Sažimanje uz ljudsko vođenje je vrlo nepraktična, te neprimjenjivo u stvarnom svijetu. 1989. godine jedan od studenata pod Barnsleyevim vodstvom, Arnaud Jacquin je dobio ideju da se slika podijeli na blokove slike (engl. *range blocks*), te se zatim pronađe lokalni iterativni funkcijski sistem (LIFS/PIFS) za svaki takav blok. Ova ideja, koju je predstavio u svojoj doktorskoj disertaciji je omogućila izgradnju fraktalnog kodera kod kojeg nije bila potrebna ljudska intervencija.

1987. godine Barnsley je zajedno s Alanom Sloanom osnovao tvrtku *Iterated Systems*. Tvrci je dodijeljeno preko 20 patenata vezano uz fraktalno sažimanje. Najpoznatija i vjerojatno najuspješnija upotreba fraktalnog sažimanja je bila još 1992. godine kada je Microsoft upotrijebio patentiranu tehnologiju zasnovanu na fraktalnom sažimanju unutar Encarta enciklopedije [10]. No, tehnologija *Iterated Systemsa* nije uspjela zaživjeti. Fraktalno sažimanje je i dalje proučavano u akademskim krugovima gdje su se na različite načine pokušali riješiti problemi vezani uz šire prihvaćanje njezine upotrebe, a neki od tih pokušaja su bili vezani i uz genetske algoritme.

Fraktalno sažimanje se zasniva na zapažanju da kod slika iz stvarnog svijeta, slično zaključku Mandelbrota, postoji određena redundancija na različitim razinama detalja. Drugim riječima, podvrgavanjem većeg dijela slike odgovarajućim linearnim transformacijama vidimo da sliči manjim dijelovima slike. Slika se podijeli na blokove slike koji se međusobno ne preklapaju, te se za svaki

takav blok pronađe odgovarajući blok veće površine koji nazivamo domenski blok, a odgovarajućom linearnom transformacijom daje zadovoljavajući rezultat u pogledu greške. Blokovi slike i domenski blokovi općenito ne moraju biti kvadrati, osim kvadrata u radovima su se koristili pravokutnici, trokuti, heksagoni, poligoni [8]. Osim samog fraktalnog sažimanja postoje radovi s hibridnim metodama, kombinacija fraktalnog kodiranja i vektorske kvantizacije [11], hibridna fraktalna DCT metoda [12], hibridna fraktalna wavelet tehnika [13].

Dekodiranje se bazira na teoremu fiksne točke. Konvencionalno dekodiranje se vrši iz proizvoljne početne slike primjenom transformacija. Nakon par iteracija tog postupka, slika konvergira prema slici koju smo kodirali.

3.1. Iterativni funkcijski sistemi (IFS)

Fraktalno sažimanje počiva na matematičkim načelima iterativnih funkcijskih sistema (IFS). Inverzni problem kojeg je definirao Barnsley se sastoji u pronalasku IFS-a za neku proizvoljnu sliku s određenom točnošću. U ovom potpoglavlju se daje matematička podloga fraktalnog sažimanja nad jednobojnim slikama.

Za preslikavanje w: $X \rightarrow X$ nad potpunim metričkim prostorom (X, d) se kaže da je kontraktivno ako postoji konstanta $0 \leq s < 1$ takva da za sve x, y iz X vrijedi:

$$\mathrm{d}\big(\mathrm{w}(\mathrm{x}), \mathrm{w}(\mathrm{y})\big) \leq \mathrm{s}\, \mathrm{d}(\mathrm{x}, \mathrm{y}) \tag{3.2}$$

Drugim riječima, međusobne udaljenosti svih točaka nakon preslikavanja ne smiju biti veće od njihove originalne udaljenosti.

Tada možemo definirati skup kontraktivnih preslikavanja w_i: $X \rightarrow X$, svaka s svojim kontraktivnim faktorima $0 \leq s_i < 1$, $i=1,\ldots, N$. Za taj skup preslikavanja definiramo faktor kontrakcije $s=max(s_i, i=1,..,N)$. Takav skup preslikavanja se naziva iterativni funkcijski sistem.

U euklidskom 2-D prostoru preslikavanje poprima oblik linearne transformacije:

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \tag{3.3}$$

Ili

$$w_i(\boldsymbol{X}) = A_i \boldsymbol{X} + T_i \tag{3.4}$$

Gdje $A_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ a $T_i = \begin{bmatrix} e_i \\ f_i \end{bmatrix}$.

A_i obavlja skaliranje, nagib i rotaciju, dok T_i predstavlja translaciju. Za slučaj da je determinanta od A_i manja od jedan preslikavanje je kontraktivno i može biti djelom IFS-a.

S *H(X)* možemo označiti skup svih neprazniih podskupova od *X*. *H(X)* se sastoji od svih crno bijelih slika, jedan element tog skupa predstavlja jedan podskup od *X*, kojeg možemo promatrati kao 2D sliku s crnom bojom na području elemenata tog podskupa te bijelom bojom kod elemenata koji nisu dio tog podskupa.

Metrika koja nam govori koliko su dva elementa *H(X)* međusobno udaljena se naziva Hausdorffova metrika.

$$h(A,B) = \max\{ supinf_{x\in A, y\in B}\, d(x,y), supinf_{x\in B, y\in A}\, d(x,y)\} \tag{3.5}$$

Hausdorffova metrika najjednostavnije rečeno označava najveću udaljenost između svih točaka iz jednog podskupa do njihovih najbližih točaka drugog podskupa. Unija preslikavanja definiranih s IFS predstavlja kontraktivno preslikavanje u metričkom prostoru *(H(X),h)* prema teoremu fiksne točke.

3.2. Teorem fiksne točke

Neka {*X*, w_i: *i=1 ,…, N*} označava IFS na potpunom metričkom prostoru *(X,d)* s kontraktivnim faktorom *s*. *W* je definiran kao *W: H(X) →H(X)* s:

$$W(B) = \bigcup_{i=1}^{N} w_i(B) : B \in H(X) \tag{3.6}$$

Tada *W* predstavlja kontraktivno preslikavanje u prostoru *(H(X), h)* , prema ranijoj definiciji kontraktivnog preslikavanja

$$h\big(W(B), W(C)\big) \leq s\, h(B, C):\ B, C \in H(X) \tag{3.7}$$

Preslikavanje *W* ima jedinstvenu fiksnu točku koja se naziva atraktor. Za atraktor vrijedi:

$$\begin{gathered} A = W(A) \\ A = \lim_{n\to\infty} W^{\circ n}(A_0),\ A_0 \in H(X) \end{gathered} \tag{3.8}$$

W^{0n} označava n-tu iteraciju uzastopnog preslikavanja.

Sljedeći primjer na smanjenoj razini ilustrira koncept fiksne točke.

Ako u jednadžbu (3.3) umjesto matrice X uvrstimo skalar dobivamo *w(x)=Ax +T*. Uvrštavajući konkretne vrijednosti *A*=0.5 i *T*=0.2 dobivamo transformaciju *w(x)*=0.5*x*+0.2. Oznaka *x* nam je nepoznanica, ali znamo da bismo primjenom transformacije željeli dobiti njega samog pa dobivamo jednadžbu *x*=0.5*x*+0.2. Očito postoji samo jedno rješenje. Ova jednadžba je rješiva direktno, no za složenije slučajeve se koristi iterativna metoda u skladu s teoremom fiksne točke. Možemo uzeti proizvoljnu početnu procjenu za *x*, primjerice *x*=20 i iterativno računati novu vrijednost na temelju jednadžbe, nakon 10 iteracija se dolazi do broja 0.41914.., dok za 15 iteracija do 0.4005981. Postupak završava kad se zadovolji neki uvjet konvergencije, ili kao što je slučaj kod fraktalnog sažimanja, nakon fiksnog broja iteracija.

Jednadžba (3.8) nam govori da će bilo koja slika uzastopnom primjenom preslikavanja definiranih s *W* konvergirati prema jednoj jedini slici. Zaključujemo kako je sliku moguće definirati samo s *W*. Za neku proizvoljnu sliku, postavlja se

problem pronalaska transformacija w_i čiji je atraktor dovoljno blizu odabranoj slici prema Hausdorffovoj metrici.

Inverzni problem je moguće riješiti s određenom greškom upotrebom teorema kolaža.

Prema njemu ako vrijedi

$$h(T, \bigcup_{i=1}^{N} w_i(T)) \leq \varepsilon \tag{3.9}$$

Onda će vrijediti i sljedeće

$$h(T, A) \leq \frac{\varepsilon}{1 - s} \tag{3.10}$$

Pronalaskom w_i takvih da imamo dovoljno malenu grešku ε, postiže se i dovoljna blizina atraktora traženoj slici T, prema izrazu (3.10).

Mogućnost sažimanja leži u pronalasku skupa preslikavanja koji se može zapisati manjim brojem bitova nego što se može nesažeta slika. Iz formule je vidljivo da će sažimanje na ovaj način biti sažimanje s gubitkom, ne postoji garancija da će atraktor biti jednak traženoj slici.

Kao što je prije bilo spomenuto, Jacquin je napravio poboljšanje metode uvodeći lokalnu karakteristiku u koncept upotrebe IFS-a. Preslikavanja w_i se ne primjenjuju na cijelu sliku, već su ograničena na određene domene slike. Prema tome, jedno preslikavanje w_i više nije specificirano samo s linearnim preslikavanjem, nego i domenom na koju se ono primjenjuje. Teorem fiksne točke i teorem kolaža vrijede i dalje.

3.3. Primjena IFS-a na slike sa sivim tonovima

U prijašnjem potpoglavlju je promatrana upotreba IFS-a nad jednobojnim slikama, dok se ovdje upotreba IFS-a proširuje na slike s sivim tonovima.

Za promatranje kako fraktalno sažeti slike sa sivim tonovima definiramo *F* kao prostor svih mogućih slika, koji se sastoji od funkcija oblika *z=f(x,y)*. Potrebno je još definirati metriku. Kod slika s sivim tonovima se mogu definirati dvije metrike.

Supremum metrika je definirana kao:

$$d_{sup}(f_1, f_2) = \sup |f_1(x, y) - f_2(x, y)| \tag{3.11}$$

Da bi preslikavanje za supremum metriku bilo kontraktivno dovoljna je kontraktivnost u smjeru intentizeta.

L2 metrika:

$$d = [\int_{S^2} [f_1(x, y) - f_2(x, y)]^2 dxdy]^{\frac{1}{2}} \tag{3.12}$$

Kod L2 metrike postoje kompliciraniji zahtjevi za kontraktivnošću. Za preslikavanje se kaže da je eventualno kontraktivno. Kontraktivnost će se postići nakon nekoliko iteracija primjena preslikavanja, ako je zadovoljen uvjet s_i<1.

W: *F*→*F* definira kontraktivnu transformaciju na metričkom prostoru slika (*F, d*), za dvije slike f_1, f_2 će vrijediti $d\big(W(f_1), W(f_2)\big) \leq sd(f_1, f_1)$, 0≤s<1.

Za takvu transformaciju *W* postoji kao i za slučaj binarnih slika, jedinstveni atraktor, slika *f* za koju vrijedi *W(f)=f*, te se do nje može doći iterativnom primjenom transformacije na bilo koju početnu sliku f_0.

Kolažev teorem za slike sa sivim tonovima glasi:

$$d\big(f_{orig}, f\big) \leq \frac{1}{1-s} d\left(f_{orig}, W\big(f_{orig}\big)\right) \tag{3.13}$$

Vidljivo je da će atraktor biti vjerniji traženoj slici što joj je sličnija njezina transformacija.

Praktična primjena fraktalnog sažimanja je sažimanje po blokovima. Slika se podijeli na nepreklapajuće blokove slike, kao i na preklapajuće domenske blokove koji zbog uvjeta kontraktivnosti moraju biti veći od blokova slike. Fraktalni kod *W* za sliku *f* se može definirati kao unija pojedinih preslikavanja za blokove. Preslikavanje se vrši nad cijelom površinom bloka slike, pa ispada da treba onoliko transformacija koliko smo blokova slike definirali.

Kako smo se prebacili u prostor više dimenzije u odnosu na binarne slike, jedna transformacija w_i sada osim preslikavanja domenskog bloka u blok slike vrši i kontrakciju intenziteta domenske slike.

$$w_i(x, y, z) = (G_i(x, y), I_i(z)) \tag{3.14}$$

G_i je geometrijski dio transformacije, *I* je dio promjene intenziteta. Geometrijski dio transformacije se sastoji od prostorne kontrakcije domenskog bloka na veličinu bloka slike, te od samog preslikavanja. Kako radimo s blokovima kvadratnog oblika, postoji 8 mogućnosti preslikavanja smanjenog domenskog bloka na površinu bloka slike. Nakon geometrijskog dijela dolazi kontraktivna promjena intenziteta piksela.

$$I_i(f) = s_i g_i(f) + t_i \tag{3.15}$$

I_i mora biti kontraktivna transformacija kako bismo mogli iskoristiti teorem fiksne točke. Transformaciju w_i možemo prikazati u matričnom obliku:

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x - D_x \\ y - D_y \\ g_i(z) \end{bmatrix} + \begin{bmatrix} R_x \\ R_y \\ t_i \end{bmatrix} \tag{3.16}$$

Oznaka g_i označava geometrijsku promjenu nad vrijednošću piksela zbog geometrijske transformacije. Parametri a_i, b_i, c_i, d_i, D_x, D_y, R_x, R_y su dio geometrijskog dijela transformacije. Parametre a_i, b_i, c_i, d_i nije potrebno određivati jer će oni biti određeni načinom preslikavanja domenskog bloka u blok slike. D_x, D_y daju informaciju o kojem se domenskom bloku radi, a R_x i R_y gdje se nalazi blok slike na koji će se domenski blok preslikati. Parametri s_i i t_i se odabiru tako da greška između transformiranog bloka i bloka slike bude što manja, u skladu s kolaževim teoremom.

Ako s R_{ij} označimo vrijednost nekog piksela na poziciji (*i, j*) unutar bloka slike, a s D_{ij} vrijednost piksela nakon geometrijske transformacije, tada grešku između dva dvodimenzionalna skupa podataka a i b možemo računati na sljedeći način. M i N predstavljaju dimenzije dvodimenzionalnog skupa podataka.

$$E = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[s_i D_{xy} + t_i - R_{xy}\right]^2 \tag{3.17}$$

Unutar formule za pogrešku vidimo postojanje dviju varijabli. Prema tome, greška će biti minimalna kada će parcijalne derivacije po s_i i t_i biti jednake 0, odnosno

$$s_i = \frac{MN\sum_{x=1}^{M}\sum_{y=1}^{N} D_{xy}R_{xy} - \left(\sum_{x=1}^{M}\sum_{y=1}^{N} R_{xy}\right)\left(\sum_{x=1}^{M}\sum_{y=1}^{N} D_{xy}\right)}{MN\sum_{x=1}^{M}\sum_{y=1}^{N} D_{xy}^2 - \left(\sum_{x=1}^{M}\sum_{y=1}^{N} D_{xy}\right)^2} \tag{3.18}$$

$$t_i = \frac{1}{MN}\left(\sum_{x=1}^{M}\sum_{y=1}^{N} R_{xy} - s_i\sum_{x=1}^{M}\sum_{y=1}^{N} D_{xy}\right) \tag{3.19}$$

Parametar s_i se ograničava na interval 0, 1 zbog zadovoljenja uvjeta kontraktivnosti. Moguće je i da će nazivnik u izrazu za s_i biti jednak 0, s_i se tada postavlja na 0. Nakon korekcija vezanih uz s_i, možemo izračunati t_i.

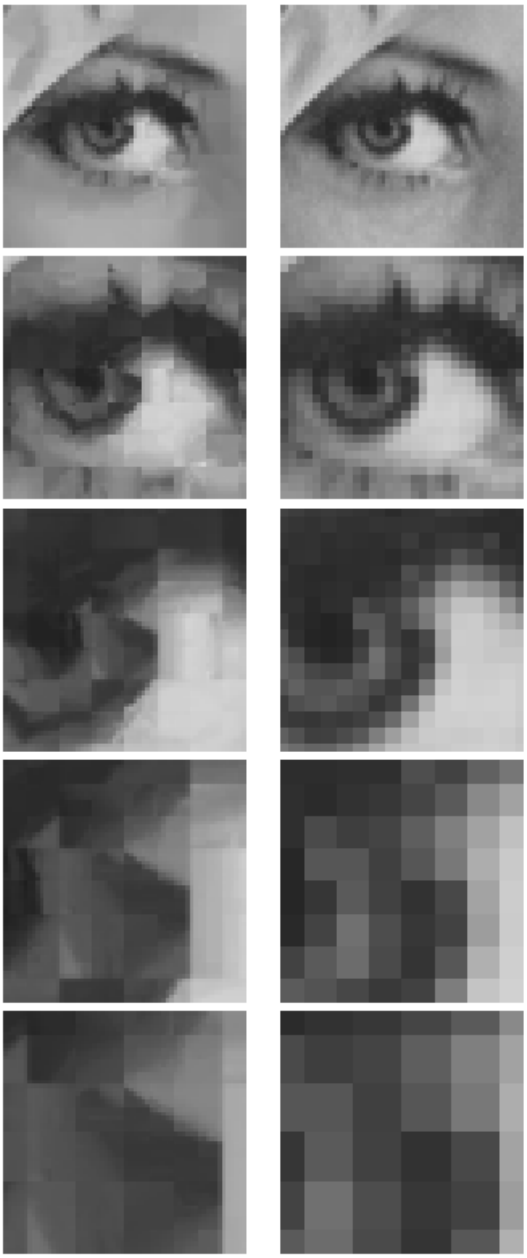
Iz dosadašnjih razmatranja možemo zaključiti kako je transformacija w_i određena s 5 parametara. Izometrijom (8 mogućnosti preslikavanja) koja se koristi kod transformacije, *x* koordinatom domenskog bloka, *y* koordinatom domenskog bloka, kontraktivnim faktorom (s_i) i pomaka u svjetlini(t_i). Koordinate R_x i R_y nisu uključene zbog toga što su tijekom kodiranja i dekodiranja one određene indeksom bloka slike tijekom procesa kodiranja. Kontraktivni faktor te pomak se moraju kvantizirati za potrebe sažimanja, te se to obavlja prije računanja MSE greške i donošenja odluke o najboljoj transformaciji.

Za konvergiranje procesa dekodiranja je ključna kontraktivnost u intenzitetu. Kontraktivnost u prostornoj domeni je pak potrebna za kreiranje detalja. Zahvaljujući kontrakciji u prostornoj domeni, početne razlike u kontrastu između blokova slike se zbog kontrakcije preslikavaju u detalje unutar blokova slike. Ako dekodiranje počnemo uniformnom slikom, nakon jedne iteracije ćemo imati sliku s uniformnim blokovima, ali različitog intenziteta (zahvaljujući preslikavanju kod intenziteta). U sljedećoj iteraciji će se razlike u intenzitetu koje postoje na

granicama blokova slike preslikavati unutar pojedinih blokova (zbog činjenice da će jedan domenski blok zbog svoje veličine prekrivati više blokova slike), čime će se stvarati detalji. Konvergencija dovoljno blizu atraktora se postiže za 8 do 10 iteracija. Postupak konvergencije prema atraktoru je vidljiv na slici 3.4.

Slika 3.4 Iterativan postupak dekodiranja, slika a) početna slika, b) rezultat nakon jedne iteracije, c) nakon 3 iteracije, d) nakon 8 iteracija

Jedna mogućnost fraktalnog sažimanja je dekodiranje slike na višim ili nižim rezolucijama. U teoriji, fraktalnim sažimanjem nisu nigdje zapisane dimenzije slika koja se sažima već transformacije. Dekodiranjem na višim rezolucijama možemo postići određenu vrstu fraktalnog zumiranja. Nedostajući detalji se stvaraju na temelju transformacija i predstavljaju fraktalnu interpolaciju. Detalji su naravno umjetni, ali zahvaljujući samopreslikavanju unutar slike koje se transformacijama ostvaruje detalji mogu biti vjerniji od obične interpolacije. Na slici 3.5 vidimo primjer fraktalnog zumiranja na višoj rezoluciji. Desni stupac predstavlja zumiranje nad originalnom slikom.



Slika 3.5 Fraktalno zumiranje

3.4. Načini podjele slike na blokove slike

Najjednostavniji način podjele slike se sastoji od dijeljena slike na blokove jednake veličine. Stupanj sažimanja će tada biti konstantan neovisno o slici koju se sažima. Očit nedostatak ovakve podjele leži u tome što se ne iskorištava činjenica da većina slika nema istu razinu detalja po cijeloj površini.

3.4.1. *Quadtree* podjela

Quadtree podjela je adaptivna podjela slike. Logično je za pretpostaviti da unutar slike postoje područja većih dimenzija nad kojima se može pronaći transformacija uz dovoljno malenu grešku. Na područjima gdje ne pronađemo dovoljno dobru transformaciju tražimo zadovoljavajuće transformacije za manje blokove. Metoda je dobila naziv *quadtree* zbog toga što se dobivena podjela dvodimenzionalnog prostora može promatrati kao stablo gdje svaki čvor može biti list ili imati 4 čvora djece. Obično se slika početno dijeli do određene dubine, te se dijeljenje za promatrani blok zaustavlja ili kada se nađe zadovoljavajuća transformacije ili kada smo došli do maksimalne dubine i samim time minimalne veličine bloka slike. Kako više veličina i pozicija bloka slike nije određena indeksom transformacije potrebno je kod zapisa transformacija dodati podatke o podjeli prostora na blokove slike. Za to je dovoljan jedan bit po odluci o spuštanju na nižu razinu. Budući je podjela na blokove adaptivna moguće je ostvariti koder varijabilnog stupnja sažimanja. Moguće su dvije opcije, koder čiji je cilj slika određene vjernosti, te koder kojem je cilj određen stupanj sažimanja.

3.4.2. Horizontalno-vertikalna podjela

Kod horizontalno-vertikalne podjele (HV podjela) se slično kao i kod *quadtree* metode blokovi slike rekurzivno dijele na manje. Razlika je u tome što se u ovoj metodi blok dijeli na dva pravokutnika, a linija podjele može biti horizontalna ili vertikalna. Mjesto na kojem se blok dijeli određuje se na temelju razlike u prosjecima dvaju uzastopnih redaka ili stupaca te udaljenosti potencijalnog mjesta podjele od ruba bloka. HV podjela pokazuje bolje rezultate od *quadtree* podjele.

3.4.3. Ostali načini podjele

Osim ovih relativno jednostavnih načina podjele razvijene su i metode podjele temeljene na poligonima. Postoje i pristupi podjele slike temeljeni na tehnici podijeli pa spoji. Jedan od pristupa koristi Delaunayeve triangulacije. Počevši od triangulacija za neki uniforman raspored točaka dolazi do koraka dijeljenja trokuta s najvećim standardnim derivacijama. Zatim dolazi korak spajanja u kojem se spajaju trokuti koji imaju slične srednje vrijednosti piksela. Kod drugog pristupa tehnici podijeli pa spoji prvo se pronađu transformacije za blokove slike najmanje veličine, te se zatim heurističkim pristupom spajaju u veće regije sve dok je transformacija primjenjiva na tako stvorenu regiju. Proučavani su i evolucijski pristupi provedbe spajanja.

4. Radovi s genetskim algoritmima i sažimanjem slika

Većina radova koja se može pronaći vezana uz genetske algoritme i područje sažimanja slika je upravo primjena genetskih algoritama nad fraktalnim sažimanjem. Način na koji je to ostvarivo može se vidjeti u sljedećem poglavlju. U ovom poglavlju je dan kratak osvrt na neke od ostalih pokušaja upotrebe genetskih algoritama kod sažimanja slika.

4.1. Genetski algoritmi i evolucija valića

U današnje doba u sažimanju slika najzanimljivije s stanovišta stupanj sažimanja/gubitak kvalitete su metode zasnovane na valićima. Performanse takvih kodera ovise o izboru dobrog valića. Koderi zasnovani na valićima uglavnom koriste standardne valiće koji su se pokazali dobrima za svakodnevne fotografije. No, u području specijaliziranih slika, kao skenirani dokumenti, otisci prsta, medicinske slike, satelitske snimke možda ne postižu rezultate kakvi bi se mogli dobiti upotrebom specijaliziranog valića. U [14] su autori upotrijebili koevolucijski GA nad „koracima dizanja“. Jedan korak dizanja kod filtara označava promjenu kojom je očuvano svojstvo inverzije. Transformirani signal se tada može savršeno rekonstruirati. Koraci dizanja predstavljaju efektivan način za konstruiranje komplementarnih filtarskih parova. Svaki valić sastavljen od konačno filtara se može predstaviti s konačno mnogo koraka dizanja. Moguće je započeti od trivijalne transformacije valićima i primjenom konačnog broja koraka dizanja doći do svih mogućih valića i pripadnih filtarskih parova koji će služiti za transformaciju. U radu su odlučili uz pomoć GA stvoriti valić za upotrebu nad slikama otisaka prstiju. Dobiveni valić su usporedili s valićem kojeg upotrebljava FBI. Evoluirani valić se pokazao boljim. Prema njihovim riječima, dobitak od 0.75dB u pogledu smanjenja PSNR se može preslikati u 15-20% manje prostora potrebnog za pohranu.

4.2. Genetski algoritmi i vektorska kvantizacija

Kao što je rečeno u jednom od prijašnjih poglavlja, ideja vektorske kvantizacije leži u promatranju skupine uzoraka kao vektor te kvantizirati takav vektor. Kvantizacija se provodi u smislu zamjene s indeksom uzorka iz rječnika koji predstavlja najmanje odstupanje od promatranog. Da bi vektorska kvantizacija bila efikasna potrebno je pronaći dobar rječnik sastavljen od određenog broja vektora koji će najbolje predstavljati sve ostale. Za dobivanje rječnika postoji više algoritama, U radovima [15], [16], [17], [18] su autori pokazali da se kombiniranjem GA i standardnih algoritama može doći do boljih rezultata nego primjenom samih standardnih algoritama.

4.3. Genetski algoritmi i evolucija filtra

Kod upotrebe diskretne kosinusne transformacije i korištenja većih stupnjeva sažimanja dolazi do deformacije u vidu jasnog razaznavanja pojedinih blokova nad kojime se vršila DCT (kod JPEG-a su ti blokovi veličine 8x8 piksela). Pojavljivanje tog efekta je posljedica kvantizacije i može se promatrati kao šum nadodan originalnoj slici. Kako bi se ta pojava ublažila može se koristiti neki filtar. Teoretski se taj šum može smanjiti upotrebom niskopropusnog filtra, ali će to za rezultat imati zaglađivanje cijele slike i gubitak detalja.

FIR filtar je opisan sljedećom jednadžbom:

$$y = \sum_{k=1}^{n} w_k x_{(k)} \tag{4.1}$$

Iz jednadžbe je vidljivo da izlaz filtra ovisi o određenom konačnom broju uzoraka ulaza. U radu [19] su koristiti L filtar te su za dobivanje težina upotrijebili GA. Težine bi se određivali za svaku sažetu sliku posebno i spremale zajedno s njom.

5. Praktični rad

U praktičnom dijelu rada su promatrana dva pristupa korištenja genetskih algoritama nad fraktalnim sažimanjem. Dok kod prvog pristupa jedinka predstavlja skup transformacija koji opisuju neko veće područje slike, kod drugog pristupa ona predstavlja jednu transformaciju. Za drugi pristup se kod sažimanja neke slike genetski algoritam izvršava za svaki blok slike. Osim razlike u prikazu jedinke postoji i razlika u metodi podjele slike na blokove. Drugi pristup koristi *quadtree* način podjele, a kod prvog se pristupa koriste dvije veličine blokova slike. Oba pristupa su generacijski genetski algoritmi s turnirskom selekcijom i koriste elitizam.

Osim samog programskog ostvarenja napravljeno je i sučelje za lakše baratanje parametrima fraktalnog sažimanja i genetskog algoritma, kao i lakšu upotrebu kodera i dekodera. Ugrađena je i podrška za slike u boji.

5.1. Usporedba slika

Dvije mjere koje se najviše koriste za usporedbu greške između slika su srednja kvadratna pogreška (engl. *Mean Squared Error,* MSE) i vršni odnos signal/šum (engl. *Peak Signal to Noise Ratio*, PSNR). Srednja kvadratna pogreška između dvije slike *f* i *g* je definirana kao:

$$MSE = \frac{1}{N}\sum_{j,k}^{N}(f(j,k) - g(j,k))^2 \tag{5.1}$$

N predstavlja ukupan broj piksela unutar slike, a *j* i *k* moguće koordinate piksela. PSNR je definiran kao:

$$PSNR = 10log_{10}(\frac{255^2}{MSE}) \tag{5.2}$$

PSNR se više koristi kao mjera greške zbog toga što je logaritamska mjera, kao što je i ljudski odaziv na intenzitet svjetlosti. Povećanjem PSNR-a povećava se i vjernost originalu. Dvije slike s PSNR iznad 40 dB su za ljudskog promatrača

jednake. No, korištenje ovih matematičkih mjera za usporedbu slika iako korisno, je istodobno i varljivo. Dvije slike dobivene sažimanjem jednakog PSNR-a u odnosu na original mogu kod ljudskog promatrača izazvati potpuno drugačiji dojam uspješnosti sažimanja. Dok za jednu sliku greška može biti raspršena po uniformnim dijelovima, kod druge u odnosu na original greška može biti koncentrirana oko rubnih područja nekog elementa slike, što će za promatrača biti vrlo uočljivo. Isto tako, možemo sve piksele slike pomaknuti u svjetlini za 10 i iako će čovjeku subjektivno slika biti poprilično istovjetna, PSNR će biti jednak ako ne i lošiji od slike dobivene JPEG-om na visokom stupnju sažimanja s vidljivim artefaktima.

Za ocjenu kvalitete načina sažimanja slike osim greške koja se tim postupkom unosi promatra se i stupanj sažimanja (engl. *Compression Ratio,* CR). Stupanj sažimanja je definiran kao omjer veličine originalne slike i slike u sažetom zapisu.

5.2. Prvi pristup fraktalnog sažimanja slike genetskim algoritmima

Prvi pristup upotrebe genetskih algoritama se bazira na dvjema razinama veličine bloka slike. Slika će biti sažeta transformacijama koje opisuju blokove veličine od *bxb* i 2*bx*2*b* piksela. Kromosom se kod ovog pristupa sastoji od niza transformacija te predstavlja jednu veću regiju unutar slike. Gena će biti koliko je transformacija potrebno za opisati sliku manjim blokovima. Transformacija ima oznaku koje je razine. Transformacija s oznakom 1 se odnosi na blok veličine *bxb*, a s oznakom 2 na blok veličine *2bx2b*. Na početku evolucije sve transformacije koje to mogu biti se postave na razinu 2. Nakon polovice generacija se počinju koristiti i transformacije razine 1. U mutaciju je ugrađen i prijelaz iz jedne razine u drugu za transformacije kod kojih je to moguće. Potreba za određenim brojem generacija u kojima se koriste samo transformacije razine 2 se pokazala jer zbog prijelaza između razina transformacija nema dovoljno vremena za konvergenciju. Ako neka transformacija druge razine prijeđe ispod određenog praga greške ona se uzima kao konačna.

U prvom pristupu se slika dijeli na regije od 256x256 piksela iz kojih se uzimaju domenski blokovi za usporedbu s blokovima slike. Korištena regija ovisi o poziciji bloka slike. Za računanje dobrote jedinke na raspolaganju su obje metrike spomenute u potpoglavlju 3.3.

5.2.1. Prikaz jedinke

Kromosom je sastavljen od niza transformacija. Jedan zapis o transformaciji se sastoji od:

- x koordinate domenskog bloka,
- y koordinate domenskog bloka,
- izometrije koju treba primijeniti,
- kontrasta,
- pomaka,
- greške koja se postiže ovom transformacijom,
- zastavice o promjeni,

- zastavice o dovoljno maloj grešci,
- razine.

Dio nad kojim radi genetski algoritam se sastoji od X koordinate, Y koordinate i izometrije. Kontrast i pomak je efikasnije odrediti deterministički metodom najmanjih kvadrata. Nakon određivanja tih paramatera mogu se odrediti i ostali elementi zapisa. Zastavica o promjeni postoji zbog ubrzavanja izvršavanja genetskog algoritma. Nepotrebno je ponovno računati grešku prilikom izračuna dobrote neke jedinke ako transformacija unutar kromosoma nije bila izložena mutaciji. Kako je vjerojatnost mutacije relativno mala, ispitivanje ove zastavice prije računanja greške predstavlja značajnu vremensku uštedu. Razina govori radi li se o bloku veličine *bxb* ili 2*b*x2*b*, ¾ transformacija unutar kromosoma će moći biti jedino razine 1, a ostala ¼ može biti i razine 2. Blokovi slike druge razine su četiri puta veći od blokova prve razine, pa je pokrivanje cijele slike moguće s ¼ ukupnog broja transformacija, te taj broj predstavlja granicu sažimanja za ovaj pristup. Kod zapisivanja u datoteku troši se 33 bita po transformaciji. 8 bitova za x koordinatu, 8 bitova za y koordinatu, 3 bita za izometriju, 5 bitova za kontrast, 7 bitova za pomak u svjetlini, 1 bit za razinu i 1 bit za klasu. U slučaju da je bit klasa postavljen u 1 radi se o transformaciji gdje je razlika u pikselima malena, pa se zapisuje samo pomak svjetline.

5.2.1.1 Dobrota jedinke

Dobrota jedinke ovisi o grešci koja se unosi sažimanjem. Spomenuto je da svaka transformacija ima pridruženu grešku. Dobrota se tada može dobiti na temelju tih grešaka i jednaka je :

$$f = \frac{1}{1 + \frac{1}{N}\sum_{i=1}^{N} err(w_i)} \tag{5.3}$$

N predstavlja broj transformacija koje opisuju sliku te nije jednak broju transformacija u kromosomu, postojat će transformacije druge razine čime će određen broj transformacija prve razine biti neaktivan. Kako je stupanj sažimanja ovisan o broju transformacija, moguće poboljšanje bi bilo uvesti parametar, koji bi zajedno s brojem transformacija utjecao na dobrotu. Podešavanjem tog parametra moglo bi se birati između većeg stupnja sažimanja ili bolje kvalitete slike.

5.2.1.2 Križanje

Prvotnim testiranjima se pokazalo kako uobičajeno križanje s jednom ili više točaka prekida ne daje zadovoljavajuće rezultate, te je bilo potrebno uvesti ciljano križanje s uzimanjem najboljih dijelova oba roditelja. Kod usporedbe transformacija različitih razina, izračuna se srednja greška na temelju grešaka transformacija niže razine za regiju koja je pokrivena transformacijom više razine drugog roditelja. Zatim se kao i kod transformacija iste razine uzima ona s manjom greškom. Ovime se prednost daje transformacijama niže razine, jer će dio slike uvijek biti lakše opisati većim brojem transformacija. No, u slučaju da je transformacija više razine ispod praga, križanjem će biti osigurano njezino pojavljivanje u djetetu.

5.2.1.3 Mutacija

Mutacija radi nad pojedinom transformaciji, nasumično se odabire promjena jednog od parametra za koje je rečeno da su pod utjecajem genetskog algoritma (X, Y koordinata, izometrija). Drugi mogući pristup mutaciji je unošenje šuma u parametar umjesto zamjene nasumičnom vrijednošću.

5.2.2. Utjecaj veličine slike

Broj blokova slike će se povećati linearno s veličinom slike, a broj generacija će biti jednak. Prema tome se za prvi pristup očekuje linearna ovisnost vremena izvođenja o površini slike. Slično će biti i za slučaj potpune pretrage, količina pretraživanja domenskih blokova po jednom bloku slike ostaje jednaka zbog fiksne površine pretrage od 256x256 piksela.

Tablica 5.1 Utjecaj veličine slike za prvi pristup

Slika	Vrijeme (s)	CR	PSNR (dB)
Lenna_128	9	5.17	24.80
Lenna_192	19	6.39	26.46
Lenna_256	35	7.28	27.14
Lenna_384	63	9.14	29.58
Lenna_512	114	10.81	31.40

Rezultati (Tablica 5.1) su dobiveni uz vjerojatnost križanja od 0.8, vjerojatnost mutacije od 0.1, 50 jedinki i 100 generacija te MSE prag postavljen na 50.

Vidljiv je porast vremena izvođenja kod kojeg postoji odstupanje od očekivanog trajanja. Povećanje stupnja sažimanja i PSNR-a se može objasniti relativno manjom razinom detalja unutar nekog bloka, s većim rezolucijama iste slike, postojat će više uniformnih područja koje je lakše kodirati. Kod manje rezolucije više blokova slike će sadržavati bridove i nagle promjene intenziteta. Naravno, ova opaska vrijedi za testnu sliku Lenne, i moguće je pronaći situacije kod kojih će ponašanje biti drugačije. Odstupanje od očekivanog vremena izvođenja možemo objasniti ranijim pronalaskom zadovoljavajućih blokova veće razine, što rezultira manjim vremenom računanja dobrote jedinke, te ukupno manjem vremenu izvođenja od očekivanog.

5.2.3. Ovisnost o minimalnoj veličini bloka slike

Veličina bloka ima najveći utjecaj na stupanj sažimanja. Uzevši 4 puta veće blokove slike, stupanj sažimanja će isto tako biti otprilike 4 puta veći. Kad bismo koristili samo jednu razinu blokova, tada bi bio točno 4 puta veći. Pad PSNR-a je očekivan jer je teže pronaći domenski blok koji bi dao dovoljno dobro preslikavanje. U tablici 5.2 su navedeni rezultati. Testna slika je lenna_512 s parametrima jednakim prošlom testiranju.

Tablica 5.2 Utjecaj minimalne veličine bloka slike za prvi pristup

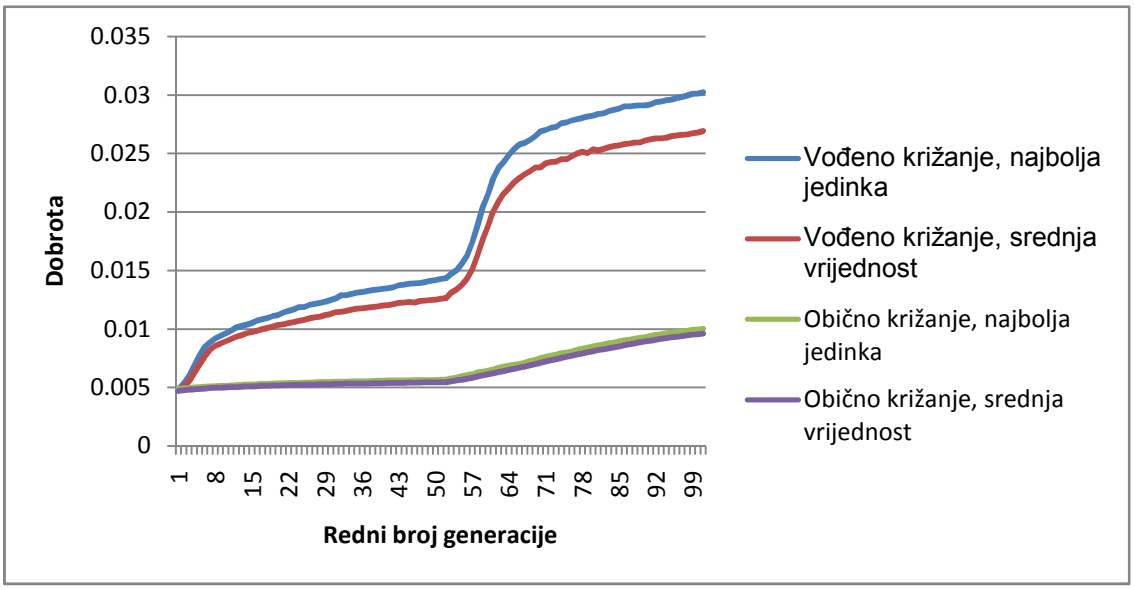
Veličina bloka	Vrijeme (s)	CR	PSNR (dB)
4x4	114	10.81	31.40
8x8	51	30.38	27.94
16x16	51	82.76	24.76

Veličina bloka utječe i na vrijeme potrebno za kodiranje.. Kod računanja greške za neki gen kromosoma, potrebno je proći kroz sve piksele bloka slike i domenskog bloka. Očito je da će zbog toga količina posla za izračun greške rasti s površinom bloka. Jedino smanjenje u konačnom vremenu će biti zbog manjih kromosoma nad kojima radi genetski algoritam. Upravo zbog toga vrijeme izvođenja ne pada linearno.

Sva ostala testiranja su rađena uz minimalnu veličinu bloka od 4x4 piksela, korištenje većih blokova rezultira lošijom subjektivnom kvalitetom iako PSNR između dvaju slika različitih veličina blokova može biti jednak.

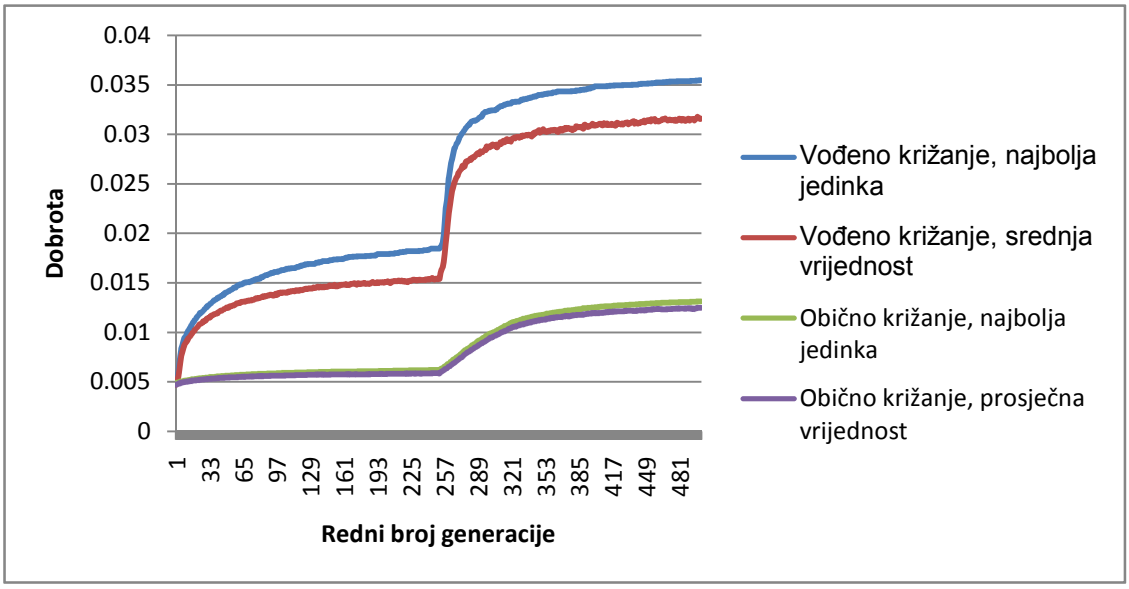
5.2.4. Utjecaj operacije križanja i broja generacija

Zbog činjenice da se ulazna slika dijeli na regije od 256x256 piksela i zatim nad tim obavlja genetski algoritam, praćenje utjecaja raznih parametara na ponašanje genetskog algoritma nad većom slikom kroz generacije bi bilo neostvarivo. Zbog toga se ovaj test radi nad zadnjim blokom slike lenne_512. Parametri koji nisu navedeni su ostali jednaki onima u prošlom testu.



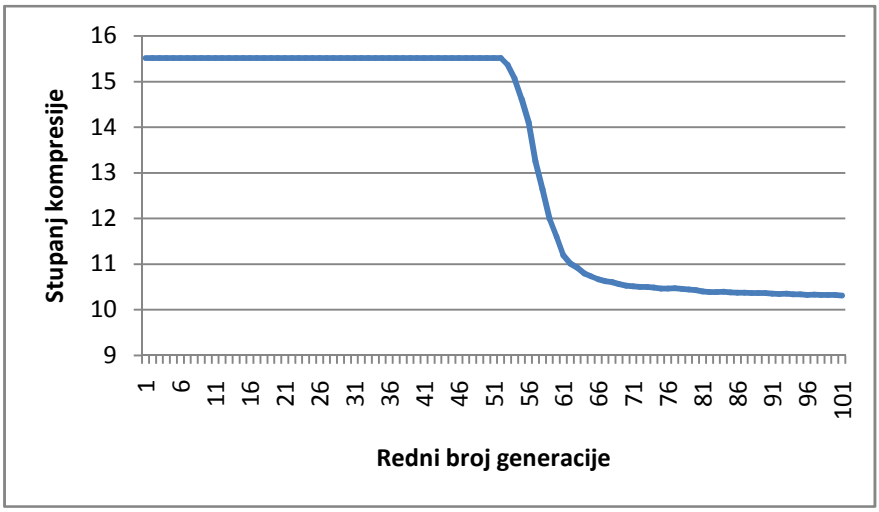
5.1 Ponašanje evolucije za križanje s jednom točkom i križanje uzimanjem najboljih dijelova oba roditelja

Iz slike 5.1 je vidljivo da se bolji rezultati postižu upotrebom modificiranog križanja. Poboljšanje dobrote koje započinje od pedesete generacije je zbog aktiviranja upotrebe manjih blokova kojima se tada detalji mogu bolje prikazati pa dobrota naglo poraste. Situacija je slična za veći broj generacija što je vidljivo na slici 5.2. Iz slika se može zaključiti da se relativno brzo dolazi do konvergencije i da je daljnji napredak kroz generacije relativno spor, što se vidi iz neznatnog dobitka u dobroti između dva testa za 50 generacija i 500 generacija.



5.2 Ponovljeno testiranje s većim brojem generacija

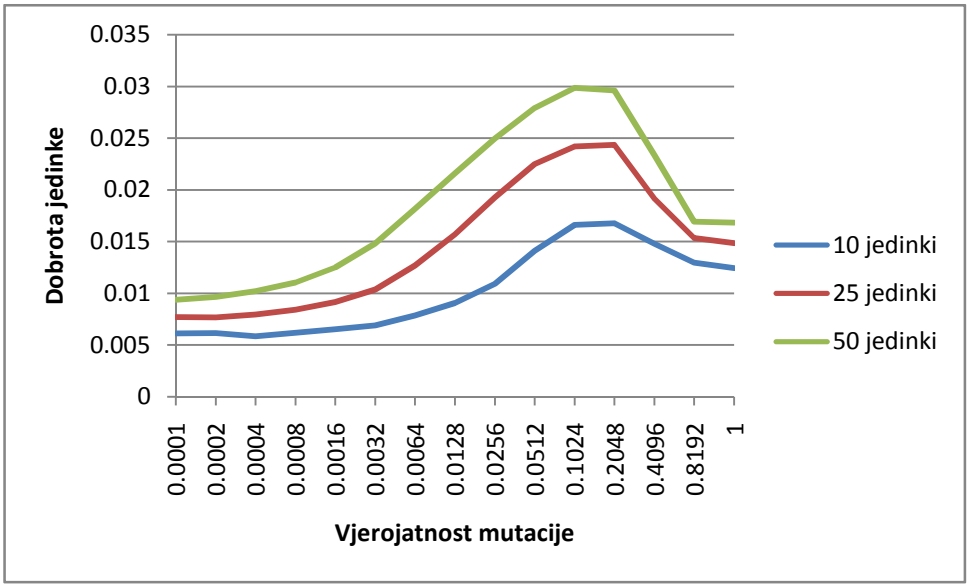
Utjecaj broja generacija na stupanj sažimanja je pozitivan, pokretanje genetskog algoritma s većim brojem generacija će rezultirati većim stupnjem sažimanja. Kako se prvih pola generacija posvećuje transformacijama više razine, u konačnom skupu transformacija koje opisuju sliku će biti veći broj transformacija više razine nego što bi to bio slučaj kod izvršavanja genetskog algoritma s manjim brojem generacija, te će stupanj sažimanja biti veći. No, unutar jednog izvršavanja, stupanj sažimanja će padati kroz generacije kao što je vidljivo na slici 5.3. Stupanj sažimanja je zapravo neznatno veći nego sto je to prikazano na grafu, kod spremanja u datoteku se za transformacije s kontrastom jednakim 0 zapisuje samo pomak u svjetlini pa je izlazna datoteka manje veličine.



5.3 Ponašanje stupnja sažimanja kroz izvršavanje genetskog algoritma

Prvu polovicu generacija genetski algoritam radi samo nad blokovima veće razine, pa je stupanj sažimanja stalan. Nakon pola generacija dolazi do upotrebe manjih blokova, čime je potrebno više transformacija za opisati sliku i stupanj sažimanja pada. Promatranjem slika 5.1 i 5.3 je vidljivo da stupanj sažimanja pada s rastom dobrote.

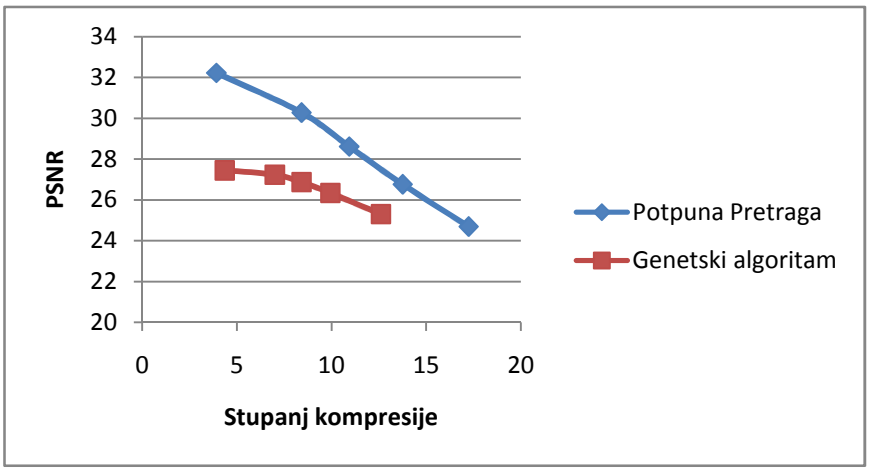
5.2.5. Utjecaj vjerojatnosti mutacije



5.4 Utjecaj mutacije

Na slici 5.4 je vidljiv utjecaj mutacije na dobrotu jedinke. Promatranjem grafa dolazimo do zaključka o potrebi za većim vjerojatnostima mutacije od konvencionalnih preporuka. Ipak, za još veće vjerojatnosti mutacije vidljiv je destruktivan utjecaj. Kao najbolji odabir se nameće upotreba vjerojatnosti mutacije između 0.1 i 0.2.

5.2.6. Usporedba s potpunom pretragom



Slika 5.5 Odnos PSNR-a i CR-a za potpunu pretragu i GA

Zbog ogromnog vremena provedbe potpune pretrage koje bi bilo potrebno nad slikom dimenzija 512x512 piksela, sljedeći test je rađen s slikom lenna_256. Na slici 5.5 se mogu vidjeti performanse genetskog algoritma u odnosu na potpunu pretragu, vidljivo je da potpuna pretraga postiže bolje rezultate, ali pogledom na tablicu 5.3 očita je drastična razlika u vremenima pretrage.

Tablica 5.3 Usporedba vremena izvođenja u sekundama genetskog algoritma i potpune pretrage za različite pragove MSE greške korištenih kod slike 6.5

Prag MSE greške	GA	Potpuna pretraga
0	56	12024
40	37	2797
100	36	1704
200	34	984
400	31	494

Razlika od 4dB koja je vidljiva na slici 5.5 za prag MSE greške od 0 je prevelika da bi se opravdalo korištenje genetskog algoritma bez obzira na dobitak u vremenu izvođenja vidljiv u tablici 5.3. Povećanjem praga MSE greške razlika između genetskog algoritma i potpune pretrage u pogledu PSNR-a se smanjuje, ali potpunom pretragom se još uvijek ostvaruje vidno bolji stupanj sažimanja.

Tablica 5.4 Rezultati sažimanja uz povećani korak pretrage uz prag MSE greške 40

Korak	Vrijeme (s)	CR	PSNR (dB)
1	3301	8.41	30.27
2	851	8.22	30.19
4	219	7.99	30.04
8	71	7.64	29.60
16	26	7.32	28.87

Korak definira veličinu prostora pretrage. Korak od 2 piksela znači pretraživanje domenskih blokova čije su obje komponente koordinate djeljive s 2.

Promatranjem rezultata pretrage za različite korake pretrage (tablica 5.4) vidimo da se za smanjenje prostora pretrage povećanjem koraka dolazi do neznatnog smanjenja stupnja sažimanja i PSNR-a, barem kad se koristi lenna_256. Dodatno povećanje stupnja sažimanja bi se moglo dobiti zapisivanjem položaja domenskog bloka manjim brojem bitova zbog većeg koraka.

Usporedbom rezultata genetskog algoritma, bez obzira na puno brže vrijeme izvođenja, dolazimo do zaključka da je bolje povećati korak pretrage nego upotrijebiti genetski algoritam. Osim povećanja koraka pretrage postoje i klasifikacijske sheme domenskih blokova koje ubrzavaju postupak sažimanja dovoljno da još više dovedu u pitanje opravdanost korištenja genetskih algoritama. Neke tehnike su opisane u [11].

5.3. Drugi pristup fraktalnog sažimanja slike genetskim algoritmima

Drugi pristup se temelji na *quadtree* metodi podjele slike. Odluka o podjeli trenutnog bloka se donosi na temelju praga. Upotreba genetskog algoritma gdje kromosom predstavlja veću regiju slike za ovaj pristup nije primjenjiva. Umjesto nad regijom slike kao kod prvog pristupa, genetski algoritam radi nad jednom transformacijom. Tekst programa vezan uz *quadtree* metodu koristi dijelove Fisherovog javnog teksta iz [11] koji su prebačeni iz C u C#.

Kako bi se ubrzala pretraga te izbjeglo stalno skaliranje domenskih blokova na veličinu blokova slike, napravljena je skalirana kopija slike. Efektivno je količina domena za pretražiti svedena na ¼ i domene koje imaju barem jednu koordinatu neparnu su ovim postupkom izuzete iz pretrage. Moguće je napraviti još tri takve kopije koje bi se podudarala za blokove s barem jednom neparnom koordinatom, ali se takva odluka pokazala neisplativom u slučaju korištenja genetskog algoritma. Dobivena slika je lošije kvalitete, a i stupanj sažimanja je manji. Lošija kvaliteta se može objasniti prostorom pretrage koji je četverostruko veći Za manji stupanj sažimanja možemo zaključiti da nije došlo do zadovoljenja praga, ali i zbog 2 dodatna bita koja je potrebno pamtiti. Kod jedne skalirane kopije gleda se svaki drugi domenski blok po X i Y koordinati pa nije potrebno pamtiti točnu koordinatu već koordinatu u skaliranoj verziji.

Za postizanje ubrzanja kod procesa izračuna greške, ne koristi se standardan način računanja MSE greške:

$$E = \frac{1}{n}\sum_{i=1}^{n}[s\,D_i + t\, - R_i]^2 \tag{5.4}$$

gdje D_i označava piksel unutar geometrijski već preslikanog domenskog bloka (skaliranje i primijenjena izometrija), s kontrast, t pomak svjetline, te R_i piksel unutar bloka slike.

Umjesto izraza (5.4) koristi se jednadžba (5.5).

$$MSE = \frac{1}{n}\left[\sum_{i=1}^{n} R_i^2 + s\left(s\sum_{i=1}^{n} D_i^2 - 2\sum_{i=1}^{n} D_i R_i + 2t\sum_{i=1}^{n} D_i\right) + o\left(nt - 2\sum_{i=1}^{n} R_i\right)\right] \quad (5.5)$$

Iako izgleda kompliciranije, promatranjem formule možemo vidjeti da sume koje se pojavljuju unutar nje već imamo izračunate za potrebe određivanja optimalnih parametara *s* i *o* promatranog bloka. Time smo smanjili količinu potrebnih operacija za računanje MSE greške, što je posebno primjetno za veće blokove slike. Kako druga metrika nema sličan, brzi način izračuna greške, u ovom se pristupu zbog potrebe za brzinom ne koristi.

5.3.1. Prikaz jedinke

U ovom pristupu je kromosom poistovjećen s jednom transformacijom. U skladu s time sastoji se od:

- X koordinate domenskog bloka,
- *Y* koordinate domenskog bloka,
- izometrije koju treba primijeniti,
- kontrasta,
- pomaka,
- MSE greška koja se postiže ovom transformacijom,
- dobrote.

Genetski algoritam je zadužen za promjene nad X koordinatom, Y koordinatom i izometrijom. Zastavica o promjeni kromosoma koja je bila zaslužna za ubrzanje izvođenja kod prvog pristupa ovdje nije uključena zbog visoke vjerojatnosti promjene djeteta u odnosu na roditelje. Pokušaj dodavanja zastavice nije pokazao značajno ubrzanje.

5.3.1.1 Križanje i mutacija

Križanje je zapravo uzimanje aritmetičke sredine. Vrijednost parametra transformacije djeteta je jednaka aritmetičkoj sredini pripadnog parametra roditelja.

Mutacijom se ostvaruje jedna od sljedećih akcija, nasumičnim odabirom:

- Dodavanja broja između [-4,4] X koordinati
- Dodavanje broja između [-4,4] Y koordinati
- Nasumična promjena izometrije

Druga ideja za mutaciju bi bila odabir jednog parametra na jednak način, ali nakon toga postupiti jednako kao s izometrijom u prethodnom načinu. Nova vrijednost bilo kojeg odabranog parametra će tada biti nasumična.

5.3.1.2 Dobrota

Dobrota jedinke je jednaka:

$$f = \frac{1}{1 + mse} \tag{5.6}$$

MSE greška se dobiva na ranije opisan način.

5.3.2. Utjecaj veličine slike

Važno je za primijetiti da kod *quadtree* metode stupanj sažimanja ovisi o dimenzijama slike. *Quadtree* metodom ćemo raditi na području cijele slike, pa će biti potreban različit broj bitova za spremanje podataka o lokaciji domene u slikama različitih dimenzija. Ovisnost je očito logaritamska. Povećanje površine slike koja se sažima za četiri puta će rezultirati potrebom za 2 dodatna bita zapisa transformacije unutar datoteke, jedan za X os, jedan za Y os.

Mjerenja su rađena s minimalnom veličinom bloka slike od 4x4 piksela i 3 *quadtree* razine, MSE pragom od 50, 50 generacija, vjerojatnost križanja 0.8, vjerojatnost mutacije 0.1 te veličinom populacije od 25 jedinki. U tablici 5.5 su navedeni rezultati testa.

Tablica 5.5 Utjecaj veličine slike za drugi pristup

Slika	Vrijeme (s)	CR	PSNR (dB)
Lenna_128	13	5.94	25.47
Lenna_192	27	6.06	27.36
Lenna_256	37	8.17	28.34
Lenna_384	79	9.37	31.13
Lenna_512	100	13.32	33.04

Uspoređujući vremena izvođenja vidimo da porast nije jednak porastu površine, ali je kao i kod prvog pristupa vidljiv porast stupnja sažimanja zahvaljujući većem udjelu transformacija više razine. Za pripadne blokove slike tih viših razina *quadtree* metodom se nije spuštalo na nižu razinu i trošilo procesorsko vrijeme.. Kako se sada pretraga vrši nad cijelom slikom, ne vrijedi linearan porast vremena izvođenja za potpunu pretragu kao što je vrijedio za prvi pristup. Osim što će se povećati broj blokova slike, povećat će se i broj domenskih blokova koje treba pretražiti za svaki blok slike, pa će ovisnost o površini slike biti kvadratna. Opažanja iz pristupa 1 vezana uz PSNR i razlog njegovog povećavanja vrijede i ovdje.

5.3.3. Ovisnost o minimalnoj veličini bloka slike

Test o utjecaju minimalne veličine bloka slike je rađen uz jednake postavke kao u prethodnom testu, testna slika je Lenna_512. Zapažanja navedena u prvom pristupu oko veličine bloka vrijede i za *quadtree* pristup. Tablica 5.6 sadrži rezultate.

Tablica 5.6 Utjecaj minimalne veličine bloka slike za drugi pristup

Veličina bloka	Vrijeme (s)	CR	PSNR (dB)
4x4	100	13.27	33.02
8x8	87	31.05	29.46
16x16	88	82.02	25.70

5.3.4. Utjecaj broja generacija

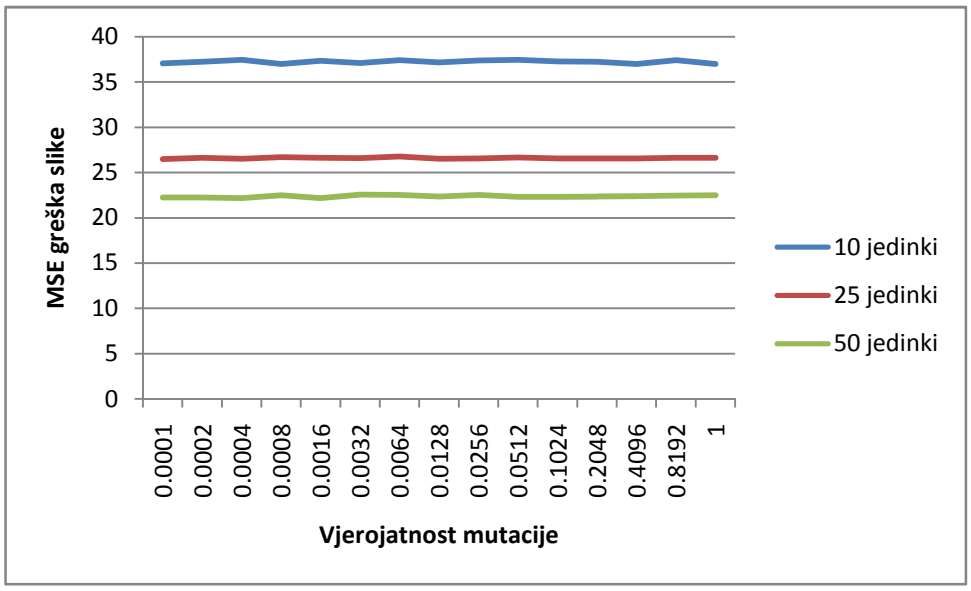
Zbog činjenice da se za svaki blok slike slijedno izvršava zaseban genetski algoritam prikazan je utjecaj broja generacija na grešku cijele slike unutar tablice 5.7, za nekoliko vrijednosti.

Tablica 5.7 Utjecaj broja generacija na CR i PSNR uz prag MSE greške od 40 (100)

Broj generacija	CR	PSNR (dB)
10	11.79(16.03)	32.75 (31.68)
25	12.29(16.76)	33.16(32.01)
50	12.41 (16.98)	33.23 (32.10)
100	12.45 (16.96)	33.28 (32.15)

Vidljiv je dolazak do zadovoljavajućih rezultata već nakon nekoliko generacija, što se nažalost može promatrati kao argument protiv upotrebe genetskih algoritama.

5.3.5. Utjecaj vjerojatnosti mutacije



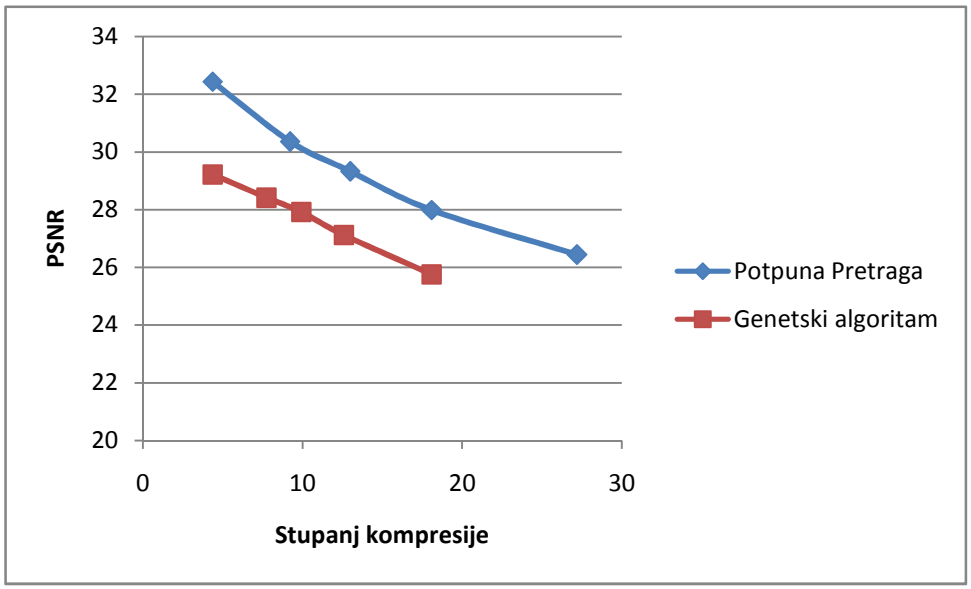
Slika 5.6 Utjecaj vjerojatnosti mutacije na grešku

Iz slike 5.6 možemo zaključiti da vjerojatnost mutacije nema nikakav utjecaj na ponašanje genetskog algoritma i kvalitetu sažimanja. Primjena križanja, mutacije ili kombinacije jednog i drugog za rezultat ima toliko različit kromosoma od prethodnog da mutacija, gledano na grešci cijele slike, ne pridonosi promjeni u

pronalaženju rješenja drugačije kvalitete. Budući se križanje obavlja uzimanjem aritmetičke sredine pojedinih komponenata transformacije, postoji određena transformacija kojoj bi se primjenom križanja sve jedinke približavale, no uz ove vrijednosti broja generacija, to se neće dogoditi i uz malene vrijednosti mutacije kao 0.001 da se to stanje izbjegne.

5.3.6. Usporedba s potpunom pretragom

Slika 5.7 prikazuje PSNR-CR odnos za potpunu pretragu i genetski algoritam. Provedeno je 50 generacija, 25 jedinki, vjerojatnost mutacije 0.1, te vjerojatnost križanja 0.8 uz pragove MSE greške 0, 40, 100, 200, 400.



Slika 5.7 Odnos stupnja sažimanja i PSNR za genetski algoritam i potpunu pretragu

Postignuti su bolji rezultati genetskog algoritma iz pogleda PSNR-CR odnosa u usporedbi s prvim pristupom, ali još uvijek nedovoljno dobri. Kao i kod prvog pristupa, povećanjem koraka pretrage je moguće postići jednake rezultate, ako ne i bolje od genetskog algoritma (tablica 5.8 i 5.9) u pogledu vremena izvođenja i kvalitete slike.

Tablica 5.8 Usporedba vremena izvođenja GA i potpune pretrage u sekundama za različite pragove greške

Prag MSE greške	Vrijeme izvođenja(GA)	Vrijeme izvođenja(PP)
0	65	2064
40	42	1430
100	33	1180
200	27	971
400	21	851

Tablica 5.9 Rezultati pretrage s različitim koracima uz MSE prag jednak 40

Korak	Vrijeme izvođenja (s)	CR	PSNR (dB)
1	1430	9.212258	30.35428
2	379	8.805	30.029
4	103	8.365	29.474
8	32	7.921	28.729
16	13	7.458	27.906

5.4. Usporedba s drugim radovima

U radu [21] upotrebljen je genetski algoritam nad jednom razinom veličine blokova. Blokovi su bili veličine 4x4 i genetski algoritam je tražio transformaciju po svakom bloku pojedinačno, slično kao u ovom radu za drugi pristup. U radu govore o postignutom PSNR-u od 36.6dB za potpunu pretragu i 31.2dB za genetski algoritam uz smanjenje vremena izvođenja za faktor 8. Kako koristi samo jednu veličinu blokova slike, stupanj sažimanja je stalan i za oba slučaja jednak 4.5. Radili su na slici Lenne dimenzija 128x128. U [20] su radili na sličan način, ali su koristili manji broj bitova za zapis jedne transformacije, te su time postigli veći stupanj sažimanja (6.73), ali manji PSNR (26.22dB) isto za 128x128 sliku Lenne, vrijeme izvođenja su smanjili za faktor 3. U ovom radu za tu dimenziju slike nisu dosegnuti ti rezultati, implementirani algoritam čak i potpunom pretragom nije u mogućnosti dosegnuti te rezultate. Razlog tome vjerojatno leži u drugačijim karakteristikama korištene testne slike budući se može pronaći različitih verzija Lenne s različitim razinama šuma. Primjenom filtra za odstranjivanje šuma mogu se dostignuti vrijednosti koje se za niske rezolucije spominju u ostalim radovima.

U [22] koriste dvije razine, s tim da iako postoji blok više razine, nad istom površinom slike mogu postojati transformacije koje bolje opisuju manje blokove podprostora većeg bloka. Na slici Lenne dimenzija 256x256 su uspjeli postići stupanj sažimanja od 10.5 uz PSNR od 30.22dB. Primjenom *quadtree* pristupa moguće je ostvariti rezultate bolje od ovdje navedenih uz pretpostavku korištenja slike Lenne s manjom količinom šuma.

5.5. Proširenje za sažimanje slika u boji

Najjednostavnije proširenje za rad s slikama u boji je provesti algoritam nad svakom komponentom (R, G, B) posebno. No, postoje određene karakteristike ljudskog oka koje se mogu iskoristiti.

Fotoreceptori mrežnice se dijele na dvije vrste, štapiće i čunjiće, oboje sadrže kemijske tvari osjetljive na svjetlost. Zbog njihove razlike u građi oni nisu jednako osjetljivi na cijeli spektar vidljive svjetlosti. Štapići su osjetljivi na svjetlo i kod niskih razina osvjetljenja, te mogu razlikovati promjene samo u osvjetljenju. Čunjići s druge strane doprinose razlikovanju boje, te postaju aktivni pri višim razinama svjetline. Brojčano gledano, štapića je 10-20 puta više nego čunjića. Iz svega toga se odavno zaključilo da je ljudsko oko osjetljivije na promjene svjetline nego promjene boje. Uzimajući u obzir konačnu gustoću fotoreceptora, može se zaključiti da oko s višim prostornim frekvencijama gubi sposobnost prepoznavanja detalja. Čunjići su u tom pogledu lošiji od štapića jer ih ima manje i rjeđe su raspoređeni.

Osnovna ideja kako iskoristiti ova saznanja u sažimanju slika je zadržati više informacija o luminantnoj komponenti i izbaciti prostorne frekvencije koje oko ne vidi. Kod JPEG-a se to razmišljanje prevodi u kvantiziranje krominancije i luminancije zasebnim kvantizacijskim tablicama, i to tako da se više frekvencije kvantiziraju većim koeficijentima, posebice za krominantne komponente.

Kako temelj fraktalnog sažimanja leži u radu nad prostornom domenom, od izbacivanja viših frekvencija nemamo koristi budući nikada ne ulazimo u frekvencijsku domenu. Preostaje nam iskoristiti poduzorkovanje krominantnih komponenata kako bismo barem djelomično iskoristili karakteristike ljudskog vizualnog sustava. U ovom je radu iskorišteno 4:2:0 poduzorkovanje. Podaci vezani uz krominantne komponente su time efektivno reducirani na ¼ originalne veličine Osim ove ideje kao mogućnost se nameće korištenje manjeg broja bitova za zapis kontrasta i pomaka, kao i korištenje manjeg broja generacija za brže kodiranje.

Za pretvorbu iz jednog prostora boja u drugi koriste se iste formule kao i za JPEG, a pretvorba se obavlja prema sljedećim formulama:

$$\begin{aligned} Y &= 0.299\,R + 0.587\,G + 0.114\,B \\ Cb &= -0.1687\,R - 0.3313\,G + 0.5\,B + 128 \\ Cr &= 0.5\,R - 0.4187\,G - 0.0813\,B + 128 \end{aligned} \tag{6.7}$$

Formule se odnose na RGB zapis gdje su uzorci boje 8bitni (256 razina), *Y* označava luminanciju, a *Cb* i *Cr* krominantne komponente.

Zaključak

U ovom je radu proučena mogućnost upotrebe genetskih algoritama u području sažimanja slika, posebice u području fraktalnog sažimanja. Fraktalno sažimanje je jedno vrijeme zaokupilo pozornost akademske zajednice, no daljnjim napretkom ostalih tehnika sažimanja, kao i zbog problema u vidu sporog kodiranja i lošijim svojstvima, ono polako gubi na popularnosti. Genetski algoritmi u fraktalnom sažimanju pomažu ako slijedimo originalnu ideju gdje se cijela slika pretražuje za najboljim domenskim blokom, opravdanje manje kvalitete slike je u bržem vremenu kodiranja. No, razvijene su metode klasifikacije koje uvelike ubrzavaju samo fraktalno sažimanje s rezultatima boljim od potrage genetskim algoritmom u istom vremenskom roku. Od dva ostvarena pristupa upotrebe genetskih algoritama, boljim se pokazao onaj koji pogledom na rezultate utjecaja mutacije u kombinaciji s osnovnom idejom više nalikuje slučajnoj pretrazi nego genetskom algoritmu. Zbog više radova koji su koristili sličan pristup očekivalo se i ponašanje sličnije genetskom algoritmu. Bolji rezultati drugog pristupa se mogu objasniti dosegom genetskog algoritma. GA je lokaliziran na jedan blok slike, pa postoji bolja kontrola greške. Kod prvog pristupa moguće je postojanje visokih grešaka na područjima rubova, kako upotrebom MSE kriterija, tako i korištenjem subjektivne ocjene greške, koje su sakrivene unutar ukupne dobrote zbog male površine takvih područja.

Iako je objavljeno dosta radova vezano uz to područje, konačan zaključak ovog rada je da u području fraktalnog sažimanja ne postoji opravdanje za korištenje genetskih algoritama. Isto tako, samo fraktalno sažimanje predstavlja zgodnu ideju, ali upotreba iste će vjerojatno ostati u području akademske zajednice.

Literatura

[1] Reeves, Colin R., Rowe, Jonathan E. *Genetic Algorithms-Principles and Perspectives-A Guide to GA Theory*. Dordrecht: Kluwer Academic Publishing, 2003

[2] Mitchell, M. *An Introduction to Genetic Algorithms*, peto izdanje. Cambridge, Massachusetts: The MIT press, 1999.

[3] De Jong, K. A. Foundations of Genetic Algorithms 2:Genetic Algorithms are NOT Function Optimizers, San Mateo Ca, USA: Morgan Kaufmann Publishers, Inc., 1993

[4] Gen, M., Cheng, R. *Genetic Algorithms and Engineering Optimization*. Kanada: John Wiley & Sons, Inc, 2000

[5] Golub, M. Genetski algoritmi-Prvi dio. Fakultet elektrotehnike i računarstva. Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave, 2004

[6] Youssef, Abdou: *Transforms in lossy compression,* http://www.seas.gwu.edu/~ayoussef/cs225/transforms.html#dctvskl

[7] Walker, J. *Wavelet-based Image Compression.* Potpoglavlje iz Transforms and Data Compression. 2000

[8] Saupe, D., Hamzaoui, R., Hartenstein H. *Fractal Image Compression An Introductory Overview.* Sveučilište Freiburg, odjel za informatiku, 1996

[9] Barnsley, M. F., Sloan, A. D. *Methods and apparatus for image compression by iterated function system,* U.S. Patent 4941193, Jul.1990.

[10] *Encyclopedia of Computer Science and Technology*, Volume 33, 1995

[11] Yuval Fisher. *Fractal Image Compression Theory and Application.* New York: Springer-Verlag, 1995

[12] Thao, N. T., *A hybrid fractal-DCT coding scheme for image compression*, in: Proc. ICIP-96 IEEE International Conference on Image Processing. Lausanne, 1996, str 169-172

[13] Duraisamy, R., Valarmathi, L., Ayyappan, J. *Iteration Free Hybrid Fractal Wavelet Image Coder.* Internation Journa of Computational Cognition, Volume 6, Issue 4, str 34-40, 2008

[14] Graseman, U., Mikkulainen, R. *Effective Image Compression using Evolved Wavelets.* Proceedings of the conference on Genetic and evolutionary computation (2005), str. 1961-1968

[15] Mohsin S., Sajjad S. *Codebook generation for Image Compression with Simple and Ordain GA.* International Journal of Computers and Communications, Volume 1, Issue 4, (2007), str. 35-40

[16] Mohammed, A.F., Al-Husainy. *A Tool for Compressiong Images Based on Genetic Algoritmhs.* Information Technology Journal, Volume 6, Issue 3, (2007), str. 457-462

[17] Pasi Franti et. al. *Genetic algorithms for large scale clustering problems.* The Computer Journal, Volume 40, str. 547-554

[18] Fernandez, V. et al. *Genetic Algorithms Applied to Clustering.* Proceedings of the International Conference on Signal an Image Processing (1996), Orlando Florida, str. 97-99

[19] Patanaik A. *Blocking Artefact Removal in Image Compression using GA.* Indian Institute of Technology, Kharagpur. Departmend of Electrical Engineering

[20] Chaprakani, Y., Soundara Rajan, K. *Genetic Algorithm Applied to Fractal Image Compression.* ARPN Journal of Engineering and Applied Sciences, Volume 4, Issue 1, 2009

[21] Xi, L., Zhang L. *A Study of Fractal Image Compression Based on an Improved Genetic Algorithm.* International Journal of Nonlinear Science, Volume 3(2007), Issue 2, str. 116-124

[22] Mitra, S. K., Murthy, C. A., Kundu, M. K. *Technique for Fractal Image Compression Using Genetic Algorithm.* IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL.7, NO.4, APRIL 1998, str. 586 – 593

Sažimanje slika upotrebom genetskih algoritama

Sažetak

U ovom diplomskom radu su istražene mogućnosti primjene genetskih algoritama na području sažimanja slika. Glavna tema unutar područja sažimanja kojom se ovaj rad bavi je fraktalno sažimanje. Dana je teoretska podloga koja stoji iza fraktalnog sažimanja, te su isprobana dva pristupa upotrebe genetskih algoritama. Ispitan je utjecaj raznih parametara na kvalitetu rješenja.

Ključne riječi:

Genetski algoritmi, fraktalno sažimanje slika

Image compression with genetic algorithms

Summary

In this diploma thesis we investigate the possibility of using genetic algorithms in image compression. The main theme of this diploma thesis is fractal compression. A theoretical background of fractal compression is given, and two different approaches are studied. The influence of different parameters on the solution quality is studied.

Keywords:

Genetic algorithms, fractal image compression

Skraćenice

CR	*Compression Ratio*	stupanj sažimanja
DPCM	*Differential Pulse-Code Modulation*	diferencijalna pulsno-kodna modulacija
GA	*Genetic Algorithm*	genetski algoritam
IFS	*Iterated Function Systems*	iterativni funkcijski sistemi
PCM	Pulse-Code Modulation	pulsno-kodna modulacija
PSNR	*Peak Signal to Noise Ratio*	vršni odnos signal/šum

Dodatak: Primjeri slika

Nesažeta slika Lenna_512

Nesažeta lenna_256

Nesažeta lenna_128

Primjer slika sažetih drugim pristupom

CR=10.037, PSNR= 32.007dB

CR=7.706, PSNR=28.93dB

CR=5.953, PSNR=26.65dB

CR=19.638, PSNR=29.788dB

CR=13.001, PSNR=27.31dB

CR=8.411, PSNR=25.23dB

Primjer slika sažetih prvim pristupom

CR=8.275, PSNR=28.835dB

CR=6.875, PSNR=27.20dB

CR=5.085, PSNR=24.75dB

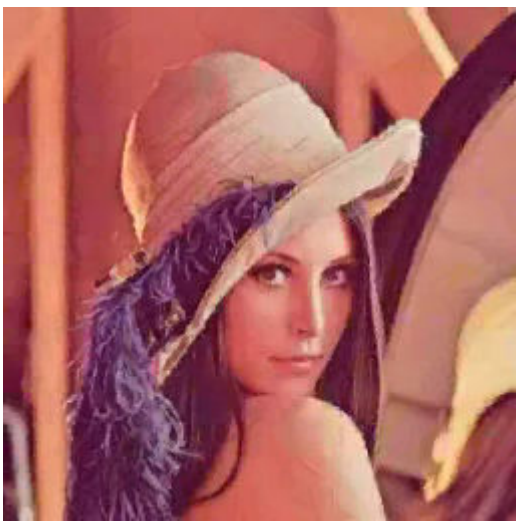
CR=11.375, PSNR=28.459dB

CR=9.97, PSNR=26.41dB

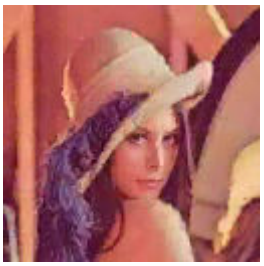
CR=6.67, PSNR=24.33dB

Primjer sažetih slika u boji drugim pristupom

CR=26.43, PSNR(Y)=31.98dB, PSNR(Cb)=33.96dB, PSNR(Cr)=33.96dB



CR=18.726, PSNR(Y)=27.87dB,
PSNR(Cb)=32.04dB, PSNR(Cr)=32.29dB



CR=13.81, PSNR(Y)=25.61dB,
PSNR(Cb)=31.12dB, PSNR(Cr)=31.33dB