УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

ФРАКТАЛНА КОМПРЕСИЈА СЛИКЕ

дипломски рад

Кандидат:
Дарко Штерн

Ментор:
проф. др **Слободан Вукосавић**

Београд, септембар 2006. године

САДРЖАЈ

1. УВОД

Намера овог дипломског рада је да сагледа једно од решења за ефикасно паковање слике у меморију ради њеног каснијег приказа. Где је проблем у складиштењу слике? Предпоставимо да је дигитална слика стандардне величини 512x512 пиксела[1]. Ако се сваки пиксел представи са 8 бита, за чување једне монохромне слике потребно нам је 262КВ, док за чување слике исте резолуције у боји потребно је 786КВ. Проблем је у томе што *слика садржи огромну количину података*! Ако ове цифре нису деловале довољно уверљиво погледајмо колико би нам меморије било потребно за смештање, рецимо, 2 сата филма када би смо чували информацију о сваком пикселу. При фреквенци освежавања екрана од 25слика/s, за меморисање филма било би потребно 141.5GB! А ово је све при резолуцији 512x512. Са повећањем величине слике заузеће меморије се пропорционално повећава. Бројеви јасно говоре да је то превише не само са становишта складиштења слике већ и брзине приказа таквих слика. Ипак, највећи проблем је при дистрибуцији, тамо где су битски протоци јако мали, рецимо као код видео секвенца мобилних телефона.

Јасно је да се морају наћи методи смањења броја бајтова потребних за складиштење слике. Компресија слике је област електротехнике која се бави овим проблемом и спада у једну од метода дигиталне обраде слике. Обзиром на велику потребу за компресијом слике, поступци за компресију последњих двадесет година нагло су се увећали. Генерално, данас се развијени алгоритми могу поделити у две велике групе, и то: алгоритми за компресију са губицима и алгоритми компресије без губитака информација о слици. Логично је да је степен компресије са губицима далеко већи, али цена у губицима информација не мора нужно бити велика. Данас развијени алгоритми са губицима који постижу степен компресије и од око 50 пута са задовољавајућем квалитетом репродуковане слике. Овим смо се у најкраћим цртама упознали са проблемом.

У наставку желимо да детаљније дефинишимо циљ овог рада, а то је да се испрограмира апликација која ће:

- покупити слику са web камере,
- компресовати слику у излазни фајл,
- декомпресовану слику приказати на екрану.

У суштини овим радом се представља један релативно нов алгоритам за компресију слике са губицима, назван *фрактална компресија слике.* Фракталан компресија се интензивно развија од средине деведесетих али још увек није стандардизована. Због тога се на интернету могу пронаћи многи алгоритми за фракталну компресију слике, а и у овом раду су предлагана различита решења.

[1] пиксел (*eng.* pixel, скраћено од picture element) је најмања јединица дигиталне слике.

Да појаснимо, фракталана компресија слике је нестандардизован метод компресије, а то значи да се слика која је компресована у излазном фајлу не може погледати у неком од стандардних програма за приказ или обраду слике, рецимо ACDSee или Photoshop. Ако се компресована слика жели приказати на монитору рачунара мора постојати посебно написан програм за њену декомпресију. Из истог разлога не постоји стандардна (или боље речено не постоји никаква) екстензија за фајл фрактално компресоване слике. У овом раду се даје екстензија излазном фајлу .frac али је то чисто из семантичних разлога.

Можда би се сада могло поставити питање зашто није коришћен неки од већ стандардизованих метода компресије? Одговор лежи у чињеници да ни један постојећи метод за сада није пружио супериорно решење проблема над осталим. Свака компресија има своје предности и своје мане, а у овом раду се управо и жели представити не само могућности постојања још једног метода компресије, већ и домени њене могуће комерцијалне примене, где би заменила постојеће алгоритме са лошијим резултатима.

Тренутно најпознатији и најраспрострањенији стандаризован метод за компресију слике је **J**oint **P**hotographic **E**xperts **G**roup (JPEG). Овај стандард нуди одличан квалитет репродуковане слике само када степен компресије иде до 20 или 25 напрема 1. Такав квалитет слике JPEG постиже одбацивањем виших хармоника из спектра слике. Међутим када су границе изнад 30:1, слике постају исувише ″блоковске″, и стога квалитет слике постаје лош за практичну примену. Такође, JPEG има проблем и са увећањем слике при истој резолуцији. Наиме, уколико се слика жели увећати, неопходно је умножавање пиксела, што опет доводи до израженог блоковског ефекта.

Управо зато, фрактална компресија покушава да пронађе своје место у системима где је потребан велики степен компресије, а да притом понуди боље решење за повећање величине слике при истој резолуцији.

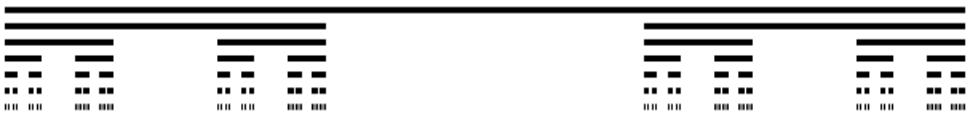
Фрактална компресија није уско везана само за слику, већ је интересантна и за нека друга подручја науке. Данас је развијена цела математичка теорија заснована на фракталној геометрији која се предаје као засебан курс на универзитетима у Америци. У овом раду ће бити изложени неки делови тог курса у виду математичког апарата фракталне компресије.

2. ДЕФИНИЦИЈА И ИСТОРИЈА ФРАКТАЛА

Шта је фрактал? Да би смо лакше разумели појам фрактала, погледајмо хронолошки како се до њега дошло.

Објекте које данас називамо фракталима откривени су далеко пре него што је појам фрактала постојао. Интересантно је да су ти објекти најчешће припадали апстрактној математичкој анализи и чинило се да они немају никаву коресподенцију у реалном свету. Данас је показано управо супротно, слике природе имају највише изражену фракталну геометрију.

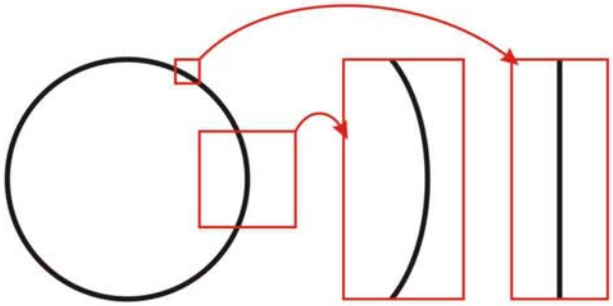
Још је Кантор[2] при испитивању особина граничних вредности скупова конструисао веома интересантан пример скупа који се добија дељењем дужи. За почетни члан низа узео је интервал [0,1], следећи члан низа формирао је тако што је од предходног низа избацио средишњу трећину, а наредни члан низа је формирао тако што је избацио средишњу трећину од интервала који су преостали у предходном низу и тако даље. Графички *Канторов скуп* се може приказати као на Сл. 2.1.



Сл. 2.1. *Кантонов скуп*

Уколико се поступак продужи у бесконачност добија се скуп који поседује особину да *непразан подскуп који припада произвољном отвореном интервалу има исту структуру као и цео скуп. ″*Самосличност*″* је једна од основних особина фрактала. Канторова прашина, како се још назива због *″*распршивања*″* интервала на тачке, је први познати пример фрактала. Појам *самосличност* заслужује мало детаљније објашњење. Сетимо се само колико је проблема било да се разуме облик земље. Данас, када постоје слике из свемира, ми јасно можемо видети да је Земља округла, али пре 500 година када људи нису могли да се отисну тако високо, површина Земља је деловала као равна плоча. Закључак о облику земљине површине се разликовао због скале (положаја) посматрања. На Сл. 2.2. а) је дата цела кружница и на овој скали посматрања се јасно види да је реч о кружници, али ако се повећа скала посматрања као на слици б) кружница постаје лук. Даљим повећањем скале посматрања добићемо праву линију као на Сл. 2.2. ц).

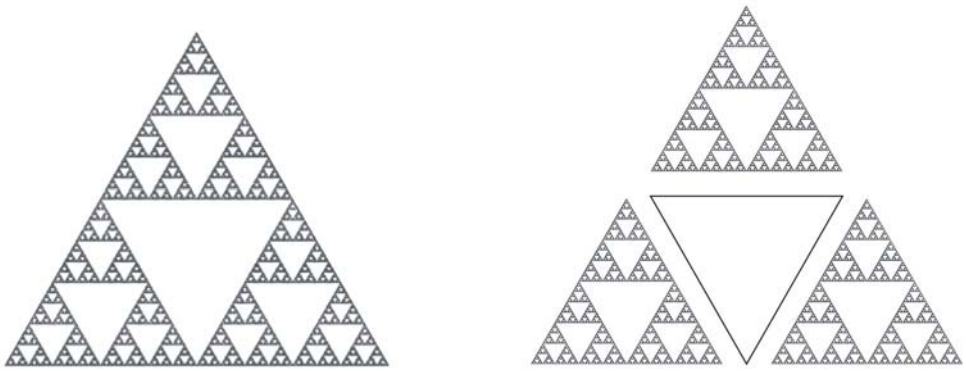
[2] Georg Ferninand Lidwig Philipp Cantor, 1845-1918



а) б) ц)

Сл. 2.2. *Три скале посматрања истог објекта*

Изглед, или тачније речено структура, класичних објеката у великој мери зависе од скале посматрања. А погледајмо сада још један можда и најпознатији пример фрактала, Сиерпинскијеб[3] троугао. На Сл. 2.3. а) дат је изглед Сиерпинскиевог троугла, док се на слици б) јасно види да је он састављен од три њему индентична трогугла. Ма који од та три троугла да разбијемо на њему мања три, они ће опет бити инденични по свом облику и структури полазном Сиерпинскиевом троуглу.



а) б)

Сл. 2.3. *Сиерпинскиев троугао*

Можемо да закључимо, ф*рактали задржавају свој облик и комплексност без обзира на скалу посматрања*, тј. не могу се разложити на једноставније елементе. Та особина се назива самосличност.

Покушајмо сада да одредимо тополошку димензију објекта Канторовог скупа. По конструкцији се види да није права, значи није димензије 1, па мора бити тачка (димензија 0), али по особинама није ни тачка. Узмимо сличан пример из свакодневног живота. Ако посматрамо површину папира и тај папир згужвамо. Колика је топлошка димензија згужваног папира? Он описује простор па би требала да је 3, али и даље је реч о површини, дакле 2. Па које је димензије фрактал? Одговор на питање дао је Хаусдорф[4], генерализацијом појма димензије објекта. У топлошком смислу димензија произвољног облика може бити само цео број. Хаусдорф је дефинисао димензију објекта на основу начина на који објекат попуњава простор. Димензија објеката који равномерно попуњавају простор биће иста по тополошкој и Хаусдорфовој дефиницији, али

[3] Waclaw Franciszek Sierpinski, 1882-1969
[4] Felix Hausdorff, 1868-1942

за објекте који показују извесну ирегуларност, попут Канторове прашине или згужваног папира, њихова димензија биће реалан број. Фрактална димензија[5] [1, 8, 9], представља посебан случај Хаусдорфове, односно формула (1) се може применити само код објеката који показују својство самосличности.

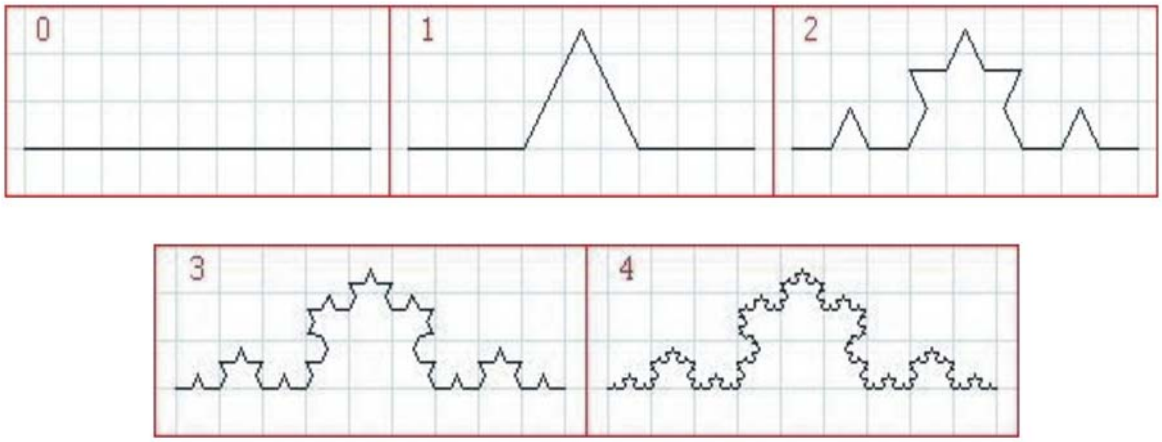
$$d = \frac{\log P}{\log S} \tag{1}$$

У формули d је фрактална димензија објекта, P број самосличних делова, а S фактор скале. Па је према формули димензија Канторове прашине

$$d = \frac{\log 2}{\log 3} \cong 0.631 \tag{2}$$

јер је број самосличних делова 2, а то су прва и трећа трећина, док је фактор скале 3, пошто постоје 3 трећине. Дакле, димензија Канторовог скупа је реалан број који је већи од топлошке димензије. Особина да је *фракталана димензија већа од тополошке*, је друга врло важна особина фрактала. Треба посебно нагласити чињеницу да фрактална димензија не мора нужно бити не целобројна вредност, али да би објекат био фрактал тај број мора бити већи од тополошке димензије. Случај који ово илуструје је Пеанова[6] крива, чија је фрактална димензија 2, а тополошка 1, обзиром да је крива. Она је такође пример да се са кривом може покрити цела површ.

На примеру конструисања Кохове[7] пахуљуце показаћемо још једну јако важну особину фрактла. Да би смо добили Кохову пахуљицу потребно је уместо средње трећине почетног сегмента конструисати једнакостраничан троугао без доње странице, а у свакој наредној итерацији поступак треба поновити над сваким праволиниским сегментом из предходне итерације, Сл. 2.4. Када се поступак продужи у бесконачност добија се *атрактор*, који представља фрактал.



Сл. 2.4. *Конструкција Кохове пахуљице*

Описана конструкција настанка атрактора, у овом сличају Кохове пахуљице, где се сваки сегмент замењује другим сегментом припада класи L-

[5] Тачна дефиниција фрактална димензије може се наћу у књизи Michael Barnsley-a ″Fractal Everywhere″, страна 174.
[6] Giuseppe Peano, 1858-1932
[7] Niels Fabian Helge von Koch, 1870-1924

систем фрактала. Овај систем измислио је Линденмајер[8], који чак није ни био математичар већ биолог, покушавајући да опише раст биљака.

Ако погледамо још једном Сл. 2.4. можемо видети и да се Кохова пахуљица може добити применом простих трансформација. Примећујемо да се наредни члан низа формира тако што се предходни члан умањи три пута, а затим се на њега надовежу још три таква иста, која су у односу на њега постављени под одговарајућим углом ($\pm\pi/3$, 0). Тако закључујемо, д*а би смо формирали атрактор, односно фрактал, можемо применити трансформације у виду итеративног процеса*. Ово је последња важна особина фрактала. Математички формирање фрактала, стога, можемо записати као унију трансформација примењених на почетни члан низа:

$$W(A) = \bigcup_{i=1}^{n} w_i(A)$$
$$A_n = W(A_{n-1}) = W^n(A_0) \qquad (3)$$

У формули (3) A_0 је почетни члан низа, A_n је n-ти члан низа, ако n тежи бесконачности добија се атрактор, односно фрактал. Фрактал добијен применом итеративних трансформација, припада тзв. класи итеративних фрактала. Систем (3) се назива *Itereted Function System – IFS* и представља основу за примену фракталне компресије слике.

Један од кључних момената за фракталну компресију лежи управо у особини да се фрактал може формирати применом итеративног процеса. Наиме, показује се да по одређеним условима за формирање фрактала није неопходно познавати почетни члан низа (A_0), односно, *формирање фрактала је једнозначно одређено само низом трансформација* W^n. Из ове чињенице, рођена је идеја фракталне компресије, где уместо да памтимо вредности сваког пиксела, довољно је запамтити само информацију о контракцији.

У наставку се поставило питање, колика је дужина атрактора? Стандардан начин израчунавања дужине криве предвиђа поделу криве на све мање сегменте док сегменти не постану праве, а дужина је једнака збиру дужина сегмената. Овај поступак, назван ректификација, када се примени на фрактале не даје резултате. Колико год мали сегмент узимали, он садржи подпуно исту комплексну структуру као и полазни фрактал, па закључујемо да фрактали нису ректифицибилни. Другим речима, фрактали имају бесконачан обим иако им је површина атрактора коначна. Управо због ове особине, да би нагласио комплексност структуре, **Манделброт**[9] их је назвао *фрактали*[10]. Реч фрактал води порекло од латинске речи fractional, што би значило делимичан, изломљен, а што се у потпуности слаже са чињеницом да су фрактали у свакој својој тачки ˝изломљени˝.

[8] Aristid Lindenmayer, 1925-1989
[9] **Benoit B. Mandelbrot,** 1924
[10] "I coined fractal from the Latin adjective ***fractus***. The corresponding Latin verb ***frangere*** means '"to break:"' to create irregular fragments. It is therefore sensible - and how appropriate for our needs! that, in addition to '"fragmented"' (as in fraction or refraction), fractus should also mean irregular, both meanings being preserved in fragment." Beriot Maldenbrot у књизи *The Fractal Geometry of Nature*, страна 4.

Манделброт се сматра ″оцем″ фракталне геометрије. Он је уочио да је варијација цене памука иста у току једног дана као и у току целог месеца. Касније је исто понашање приметио и са шумом у телекомуникационим системима. Наиме, уочио је да однос интервала у којима нема шума независтан од времена и интервала посматрања. И пре Манделброта, као што смо напоменули, математичари су примећивали да постоје примери када особине објеката не зависне од скале посматрања, али њих је тек Манделброт међусобно повезао. Своја запажања изнео је у књизи ″The Fractal Geometry of Nature″ (Фракталана геометрија природе) коју је објавио 1977. године. Ова књига донеће један нов угао сагледавања проблема у многим гранама науке.

Чак и данас, 30 година након објављивања књиге, не може се рећи да је фрактална геометрија сасвим сагледана. Чињеница која ово подупире је да не постоји опште прихваћена дефиниција самог фрактала. Најчешће коришћена дефиниција је управо она коју је дао Манделброт.

Дефиниција фрактала:
Фрактали су геометриске структуре чија је фрактална димензија већа него топлошка димезија.

Ипак фрактали се најчешће дескриптивно описују наводећи њихове три основне особине:

Особине фрактала:
1. *сличност самом себи*
2. *(фрактална) димензија је рационалан број*
3. *формирање итерацијом*

2.1. Историја фракталне компресије

Манделброт о фракталима није размишљао као о потенцијалној компресији података. Његова размишљања изложена у књизи ″Фрактална геометрија природе″ ишла су више у смеру описивања и предвиђања појава. Он је тако показао да објекти из природе, попут облака, дрвећа, планина... могу лако бити моделовани фракталном геометријом. Први који је препознао потенцијал фрактала за компресију слике је Михаел Барнслеј[11]. Он је у својој књизи ″Fractal Everywhere″ (1988) [1] развио теорију за формирање слике дајући јој име *Iteration Function Systems (IFS)*, која се у великој мери ослањала на рад Хутчинсона[12] из 1981, *Iteration Function Theory*.

Проблем генерисања атрактора, или слике, када су коефицијенти трансформације познати назива се директан проблем. Али, ако се жели постојећа слика компресовати на само пар коефицијената трансформације потребно је решити и *инверзан проблем* тј. налажење коефицијената трансформације чији је атрактор произвољна слика. Тада би се *слика памтила само као низ једноставних трансформација, и могла би се на жељеном месту лако реконструисати*.

[11] Michael Fielding Barnsley
[12] John Hutchinson

У циљу решавања инверзног проблема Барнслеј је у поменутој књизи изложио Колажну теорему (*The Collage Theorem*). Теорема је добила име због утиска да се делови слике добијени применом појединачних трансформација ″лепе″ у целину попут колажа. Решење које је Барнслеј нудио предвиђао је да се за сваку произвољну слику пронађе низ тренсформација чији ће се атрактор прихватљиво мало разликовати од оригиналног. Када се пронађе такав низ дат као у формули (3), степен компресије би износио невероватних 10,000:1. Нажалост, директно пронаћи низ трансформација за целу слику показао се као узалудан посао, који је брзо добио подругљив назив ″Graduate Student Algorithm″[13], чиме се желела нагласити његова непрактичност.

Али, управо један Барнслејов посдипломац дао је епилог причи. Наиме, Колажна теорема заснивала се на тражењу сличности између делова слике са целом сликом, што даје задовољавајуће резултате само када је реч о слици фрактала, а не и о произвољној слици. Џекуин[14] је учинио искорак предложивши да се уместо тражења сличности између целе слике са деловима, тражи сличност *већих са мањим деловима слике*. Овим је цео поступак аутоматизован и назива се *Partitioned Iteration Function Systems (PIFS)*. Своје решење изложио је у виду докторске дисертације заједно са софтвером који обавља *фракталну компресију слике*. Модификације овог алгоритма које су уследиле биле су искључиво по питању оптимизације брзине компресије и меморијске подршке.

У почетним софтверским решењима време фракталне компресије се рачунало у минутима, наравно то је било неприхватљиво. Уследили су нови алгоритми у којима је оптимизовано трагање за сличностима мањих и већих делова слике. Алгоритам који је најдаље отишао је **F**ast **F**ractal **I**mage **C**ompression (FFIC) [10, 13] и његово време компресије је брже за 25 до 400 пута од осталих алгоритама. FFIC алгоритмом брзина фракталне компресије се изједначила са брзином JPEG-а, чиме су се врата апликација са фракталном компресијом за гледање видео секвенце у реалном времену широм отворила.

[13] у слободном преводу би могло звучати, алгоритам за дипломце. Алгоритам се заснива на принципу: ухватити дипломца и закључај у лабораторији док не пронађе довољно добар IFS.
[14] Arnaud Jacquin

3. ТЕОРИЈА ФРАКТАЛНЕ КОМПРЕСИЈЕ СЛИКЕ

Математички апарат фракталне компресије слике изложен у овој глави нема за циљ само сагледавање фракталне геометрије, већ и пружање математичке основе неопходне за разумевање фракталне компресије[15].

3.1. Метрички простор

Да би се могло приступити проучавању апстактних објеката, њихових граничних вредности и услова конвергенције, почетком 20. века јавила се потреба да се класична анализа прошири и на скупове које не чине само реални бројеви. То је довело до стварања опште теорије коју називамо *функционална анализа*. У класичној анализи презентација реалног броја је могла бити нумеричка и геометријска. Просторни распоред, као геометриска презентација, се најбоље огледао у појму растојања међу реалним бројевима, па се за шире засновану анализу управо и пошло од појма растојања међу објектима неког апстрактног скупа. Као резултат уопштавања дошло се до појма *метричког простора* [1, 2, 3, 8, 9], који је за функционалну анализу оно што је скуп реалних бројева за класичну. Дефинишимо за почетак сам простор како би нам касније друге дефиниције дошле природније.

Дефиниција 1. *(Дефиниција простора)* Простор X је скуп. Тачке простора су елементи тог скупа.

Као што видимо простор је дефинисан доста уопштено, али нас занима само онај простор у коме се може мерити растојање између два објекта.

Дефиниција 2. *(Дефиниција метричког простора)* Нека је X непразан скуп и нека је $d : X \times X \to \Re$ пресликавање које задовољава следеће услове за произвољне елементе A, B, C скупа X:

- $d(A,B) \geq 0$ - позитивност;
- $d(A,B) = 0 \Leftrightarrow A = B$ - идентичност;
- $d(A,B) = d(B,A)$ - симетричност;
- $d(A,B) \leq d(A,C) + d(C,B)$ - неједнакост троугла.

Тада уређени пар (X, d) зовемо **метрички простор**, а за пресликавање d кажемо да је његова **метрика**.

Прве три особине често се називају још и аксиомама метричког простора. Из наведене дефиниције се може видети да нам није битна природа метрике, све док она задовољава горње четири особине. Користећи се дефиницијом 2,

[15] У колико се жели потпунија слика или додатно објашњење читалац се упућује на књигу Michael Barnseley-a ″Fractal Everywhere″ у којој је изложена теорија фракталне геометрије.

ми смо сада у могућности да ″меримо″ растојање између два апстрактна објекта.

Приметимо такође да најкраће растојање између два елемента скупа, односно две тачке простора, зависи од метрике. Узмимо прост пример, Еуклидском метриком можемо дефинисати најкраће растојање за раван, али ако желимо најкраће растојање у метричком простору за сферу онда метриком је потребно дефинисати кружницу.

Дефиниција 3. *(Дефиниција конвергенције)* Нека је $\{A_n\}_{n\in\mathbb{N}}$ низ тачака метричког простора (X, d) и нека је $A \in X$ произвољна тачка. Ако важи:

$$(\forall \varepsilon > 0)(\exists n_0)(\forall n > n_0)\ d(A_n, A) < \varepsilon \tag{4}$$

онда кажемо да низ $\{A_n\}_{n\in\mathbb{N}}$ конвергира по метрици d ка тачки A и скраћено записујемо:

$$\lim_{n\to+\infty} A_n = A. \tag{5}$$

Уколико гранична вредност постоји унутар скупа X, низ $\{A_n\}_{n\in\mathbb{N}}$ је конвергентан. Гранична вредност низа је јединствена.

Дефиниција 4. *(Кошијев критеријум)* Нека је $\{A_n\}_{n\in\mathbb{N}}$ низ тачака метричког простора (X, d) такав да за свако $\varepsilon > 0$ постоји природан број n_0 за који важи:

$$m > n > n_0 \Rightarrow d(A_m, A_n) < \varepsilon \tag{6}$$

тада за низ $\{A_n\}_{n\in\mathbb{N}}$ кажемо да је Кошијев.

Дефиниција 5. *(Дефиниција комплетности)* Метрички простор (X, d) у коме је сваки Кошијев низ конвергентан кажемо да је комплетан.

Повежимо сада изложену теорију са дигиталном сликом. Простор X би свакако био скуп слика одређене димензије. Остаје да се дефинише метрика над овим скупом. Најпогоднije је користити *Еуклидску метрику* дефинисану формулом:

$$d_E(A, B) = \sum_{i=1}^{N} \left[A(i) - B(i)\right]^2. \tag{7}$$

У формули (7) са d_E означено је да је реч баш о Еуклидској метрици, A и B су слике које су представљене као низ вредности својих пиксела, а $A(i)$ и $B(i)$ су вредности пиксела слике на i-том месту. Може се показати да је овако дефинисан метрички простор комплетан.

3.2. Банахов став

Тако долазимо до кључног момента у теорији фракталне компресије. У уводном делу, истакли смо чињеницу да је облик фрактала у потпуности дефинисан само трансформацијама којима се добија, независно од почетног члана. Докажимо ову тврдњу.

Дефиниција 6. *(Дефиниција контракције)* Нека је (X, d) метрички простор и $W : X \to X$ оператор за који постоји реалан број s $(0 < s < 1)$ такав да је испуњен услов:

$$d(W(A), W(B)) \le s \cdot d(A, B) \qquad \forall A, B \in X . \tag{8}$$

Тада кажемо да је W контракција метричког простора (X, d), а s зовемо фактор контракције.

Из последње дефиниције се види да је растојање пресликаних елемената мањи од почетног растојања. Ово је врло важна чињеница за даљу анализу. Примера ради, ако применимо контракцију на круг пречника 1 он ће се пресликати у круг чији ће пречник бити мањи од 1.

Дефиниција 7. *(Дефиниција непокретне тачке)* Нека је (X, d) комплетан метрички простор. Тачка $A \in X$ је непокретна или фиксна тачка контракције $W : X \to X$ ако важи:

$$A = W(A) . \tag{9}$$

За тачку A кажемо да је инваријантна у односу на функцију W.

Из предходне две дефиниције наслућујемо да итеративном применом оператора (односно трансформације) W низ ће конвергирати ка непокретној тачки. Управо та тврдња исказана је у Банаховом ставу.

Теорема 1. *(**Банахов став**)* Нека је W контракција комплетног метричког простора (X, d). Тада оператор $W : X \to X$ има тачно једну непокретну (фиксну) тачку.

Математички записан Банахов став:

$$(\exists_1 A)(\forall A_0) \lim_{n\to\infty} W^n(A_0) = A . \tag{10}$$

Доказ:

Формирајмо низ за $\{A_n\}$.

$$\begin{aligned} A_1 &= W(A_0), \\ A_2 &= W(A_1), \\ &\vdots \\ A_n &= W(A_{n-1}). \end{aligned} \tag{11}$$

На основу дефиниције 6. и (11) имамо да је:

$$d(A_{m+n}, A_n) = d\left[W(A_{m+n-1}), W(A_{n-1})\right] \le s \cdot d(A_{m+n-1}, A_{n-1}) \tag{12}$$

Ако поступак (12) продужимо, имамо:

$$\begin{aligned} d(A_{m+n}, A_n) &\le s \cdot d(A_{m+n-1}, A_{n-1}) \\ &\le s^2 \cdot d(A_{m+n-2}, A_{n-2}) \\ &\vdots \\ &\le s^n \cdot d(A_m, A_0) \end{aligned} \tag{13}$$

Применом неједнакости троугла из дефиниције 2. на $d(A_m, A_0)$ добијамо:

$$d(A_m, A_0) \le d(A_m, A_{m-1}) + \cdots + d(A_2, A_1) + d(A_1, A_0)\,. \tag{14}$$

Заменом (14) у (13) добија се:

$$d(A_{m-n}, A_n) \le s^n \cdot \left[d(A_m, A_{m-1}) + \cdots + d(A_2, A_1) + d(A_1, A_0)\right]. \tag{15}$$

Применом (12) на сваки члан у загради довољан број пута да се добије $d(A_1, A_0)$, дакле пишемо:

$$\begin{aligned} d(A_{m-n}, A_n) &\le s^n \cdot \left[s^{m-1} d(A_1, A_0) + \cdots + s \cdot d(A_1, A_0) + d(A_1, A_0)\right] \\ &= s^n \cdot d(A_1, A_0) \cdot \sum_{i=0}^{m-1} s^i \\ &= s^n \frac{1 - s^m}{1 - s} s(A_1, A_0). \end{aligned} \tag{16}$$

Ако сада пустимо да $m \to \infty$ и имајући у виду $s \in [0,1)$ добијамо:

$$d(A_\infty, A_0) \le \frac{s^n}{1 - s} d(A_1, A_0)\,. \tag{17}$$

Пошто је на основу предпоставке $\lim_{n\to\infty} s^n = 0$ и $\dfrac{d(A_1, A_0)}{1 - s}$ ненегативна константа, из (17) можемо закључити да за свако $\varepsilon > 0$ постоји $n_0 \in \aleph$ такво да важи:

$$(\forall \varepsilon > 0)(\exists n_0 \in \aleph)\ \ d(A_\infty, A_n) \le \frac{s^n}{1 - s} d(A_1, A_0) < \varepsilon\,. \tag{18}$$

Дакле низ $\{A_n\}_{n\in\aleph}$ је Кошијев, а због комплетности метричког простора (X, d) мора конвергирати ка некој тачки A, тј.

$$\lim_{n\to+\infty} d(A_n, A) = 0 \tag{19}$$

односно да је $A_\infty = A$. Ово можемо записати и мало друкчије имајући у виду почетни низ (11):

$$\lim_{n\to+\infty} A_n = \lim_{n\to+\infty} W^n(A_0) = A \tag{20}$$

Докажимо сада да је тачка A, у коју низ $\{A_n\}_{n\in\aleph}$ конвергира, непокретна тачка оператора W.

$$\begin{aligned} W(A) &= W(\lim_{n\to+\infty} A_n) \\ &= W\left(\lim_{n\to+\infty} W^n(A_0)\right) \\ &= \lim_{n\to+\infty} W^{n+1}(A_0) \\ W(A) &= A\,. \end{aligned} \tag{21}$$

Остаје још само да се покаже јединственост непокретне тачке. Предпоставимо да постоји тачка A' која је такође непокретна тачка оператора W. Пошто је:

$$d(A,A') = d(W(A),W(A')) \le s \cdot d(A,A') \tag{22}$$

па мора, због $s \in [0,1)$, бити $d(A, A') = 0$, дакле $A = A'$. Тиме смо показали да је A једина непокретна тачка оператора W.

Овим је доказ завршен.

Дефиниција 8. *(Дефиниција атрактора)* Непокретну тачку дефинисану Банаховим ставом називамо атрактор.

Уколико би пресликавање W било дефинисано као унија појединачних трансформација, математички записано:

$$W(A) = \bigcup_{i=1}^{n} w_i(A) \tag{23}$$

потребан услов да би W била контракција је да све појединачне трансформације w_i буду контракције. Уколико је $s = \max(s_1, s_2, \ldots, s_n) < 1$, где је s_i фактор контракције трансформације w_i, онда ће Банахов став важити и за овако дефинисано пресликавање W.

У доказу Банаховог става користили смо се произвољном метриком d над комплетним скупом X, јасно нам је да ће доказ важити и на скупу слика са Еуклидском метриком. Банахов став тврди да се *атрактор, тј. гранична вредност низа слика, може добити познавањем само низа контракција, а за почетни члан низа се може узети свака, произвољно изабрана, слика*. У овој чињеници лежи основа фракталне компресије, јер да би формирали слику више нам није потребно да познајемо вредност сваког пиксела већ је слика једнозначно одређена информацијом о контракцији. Проблем који се намеће је *како пронаћи контракцију чији је атрактор слика која се жели компресовати.*

3.3. Колажна теорема

Одређивање пресликавања W када нам је познат атрактор, односно слика, представља централни проблем фрактална компресије слике. Овај проблем у литератури је познат под називом инверзан проблем. Барнслеј је понудио решење овог проблема са Колажном теоремом. Он је кренуо од Банаховог става који каже да је атрактор A инваријантан у односу на контракцију W којом је добијен, тј. $W(A) = A$. Затим је доказао да ако (некако) нађемо контракцију W' за коју важи да је $W'(A) = A'$, где се A' и A прихватљиво мало разликују, тада ће се и атрактор контакције W' мало разликовати од A. Другојачије речено, ми не тражимо контракцију у односу на коју је слика инваријантна, већ контракцију у односу на коју је слика *приближно инваријантна*. На тај начин ми нећемо добити оригиналну слику, већ само слику која се прихватљиво мало разликује од оригиналне. Сада нам је јасно зашто *овај метод компресије спада у групу алгоритама који уносе губитке при компресији*.

Теорема 2. (Колажна теорема – Collage theorem) Нека је (X, d) комплетан метрички простор, $A \in X$ произвољан атрактор и нека је дато произвољно $\varepsilon \geq 0$. За изабрану контракцију $W = \bigcup_{i=1}^{n} w_i$, са фактором контракције s $(0 < s < 1)$ и непокретном тачком A_∞, тако да важи:

$$d\left[A, W(A)\right] < \varepsilon, \tag{24}$$

онда је

$$d(A, A_\infty) \leq \frac{\varepsilon}{1-s}. \tag{25}$$

Доказ:

Према дефиници непокретне тачке и Банаховог става имамо да је:

$$A_\infty = W(A_\infty) = \lim_{n \to +\infty} W^n(A_0). \tag{26}$$

Где је са A_0 означен произвољан елемент скупа X, и према Банаховом ставу непокретна тачка A_∞ је независна од њега. Ако узмемо да је $A = A_0$ одредимо растојање између A и A_∞.

$$\begin{aligned} d(A, A_\infty) &= d\left[A, \lim_{n \to +\infty} W^n(A)\right] \\ &= \lim_{n \to +\infty} d\left[A, W^n(A)\right]. \end{aligned} \tag{27}$$

Из неједнакости троуглова може се писати:

$$d\left[A,W^{n}(A)\right]\le d\left[A,W(A)\right]+\cdots+d\left[W^{n-1}(A),W^{n}(A)\right]\ \Rightarrow$$
$$d\left[A,W^{n}(A)\right]\le\sum_{i=1}^{n}d\left[W^{i-1}(A),\,W^{i}(A)\right]. \tag{28}$$

Ако сада на (28) применимо (13) добија се:

$$d\left[A,W^{n}(A)\right]\le\sum_{i=1}^{n}s^{i-1}\cdot d\left[A,\,W(A)\right]. \tag{29}$$

Заменом (27) у неједнакост (29) добија се тражени израз:

$$\begin{aligned} d(A,A_{\infty}) &\le \lim_{n\to+\infty}\sum_{i=1}^{n}s^{i-1}\cdot d\left[A,W(A)\right]\\ &= d\left[A,W(A)\right]\cdot\lim_{n\to+\infty}\sum_{i=1}^{n}s^{i-1}\\ &= d\left[A,W(A)\right]\cdot\lim_{n\to+\infty}\frac{1-s^{n}}{1-s}\\ &= d\left[A,W(A)\right]\cdot\frac{1}{1-s} \end{aligned} \tag{30}$$

односно

$$d(A,A_{\infty})\le\frac{\varepsilon}{1-s}. \tag{31}$$

Чиме је доказ завршен.

Колажна теорема показује да уколико пронађемо низ трансформација таквих да свака трансформација пресликава довољно близу (ε) целу слику у један њен део тада ће унија тих транформација, када се примени велики број пута над произвољном сликом, као резултат дати слику која се прихватљиво мало $\left(\frac{\varepsilon}{1-s}\right)$ разликује од оригиналне слике.

Барнслеј и Слоан[16] су предложили алгоритам компресије слике који је предвиђао употребу Еуклидске метрике, а за контракционе трансформације w_i усвојили су *афине трансформације*. Овај алгоритам, сажето изложен у предходном пасусу, није давао задовољавајуће резултате, јер је захтевао велику људску асистенцију. Џекуин је цео процес компјутерски аутоматизовао, тиме што је Колажну теорему применио не над целом сликом већ на делове слике. Пресликавањем већих делова слике у мање, експлатишу се локалне сличности, које су јако заступљене на реалним сликама.

Изведимо сада један јако важан закључак. Ако се осврнемо још једном на Банахов став и формулу (3), можемо приметити је да величина слике добијене трансформацијама $W(A)$, одређена само величином почетне слике A_0. Како смо према Коложној теореми рекли да се за почетну слику може изабрати слика прозвољног садржаја и величине, закључак је да *величина декомпресоване слике не зависи од величине слике чија се компресија врши*. Ова чињеница је

[16] Alen Sloan

сасвим у сагласности са идејом фрактална компресије да се садржај слике не памти, већ да се слика описује низом трансформација који је независтан од скале посматрања.

3.4. Афине трансформације

На предходним страницама није нас занимало какве су трансформације w_i све док су оне контракције. У алгоритму компресије потребно је упоређивати делове слике са сликама добијеним применом транформација како би се пронашли оптимални коефицијенти тих трансформације. Зато нам је потребно да познајемо и облик трансформација. Најпогодније за коришћене су афине трансформације.

Дефиниција 9. *(Афине трансформације)* Трансформација $w: R^2 \to R^2$ у облику

$$w(x_1, x_2) = (ax_1 + bx_2 + e,\ \ cx_1 + dx_2 + f) \tag{31}$$

где су a, b, c, d, e и f реални бројеви, назива се (дводмензионе) афине трансформације.

Афине трансформације су линеарно пресликавање и ми ћемо их чешће писати у векторском облику:

$$w\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{32}$$

Афиним трансформацијама могуће је вршити ротирање, скалирање и транслирање вектора у простору. Ако су x и y координате положаја пиксела слике, онда би се трећа координата z у фракталној компресији могла користити за његову боју, тако да се неки коефицијенти матрица у (32) анулирају:

$$w\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix} \tag{33}$$

Односно,

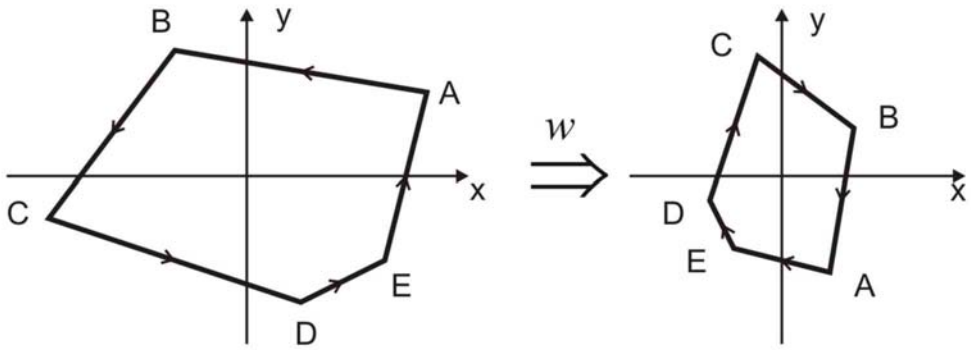
$$w\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} ax + by + e \\ cx + dx + f \\ sz + o \end{bmatrix} \tag{34}$$

Из векторског облика транформације w дат са (34) види се да се *независно* трансформишу координате и боја. Са коефицијентом трансформације s означен је *контраст* (eng. scale factor), а коефицијентом o *осветљење* (eng. offset). Коефицијенти e и f врше транслацију по x, односно y оси, док се са коефицијентима a, b, c и d врши скалирање и ротација.

Уобичајен начин за запис коефицијената a, b, c и d је тригонометриски па се може писати:

$$w\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} \rho\cos\varphi & -\rho\sin\varphi \\ \rho\sin\varphi & \rho\cos\varphi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{35}$$

У (35) φ представља угао ротације, а ρ коефицијент скалирања. Уколико је ρ негативно фигура која се пресликава мења орјентацију, Сл. 3.1.



Сл. 3.1. *Пресликавање афиним трансформацијама*

Да би смо користили афине транформације као трансформације w_i, оне морају задовољавати услов контракције. Запишимо афине транформације дате са (33) у нешто једноставнијој нотацији:

$$w(\mathbf{X}) = \mathbf{A} \cdot \mathbf{X} + \mathbf{B} \tag{36}$$

Без жеље за дубље залажење у теорију алгебре, дефинишимо сопствену вредност оператора $\mathbf{A}$ као нуле полинома $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. Да би трансформација w била контракција, потребно је да скаларни фактор сопствене вредности буде мањи од 1 или математички записано:

$$\sqrt{|\lambda|} < 1. \tag{37}$$

Ако израчунамо λ, сопствену вредност операта $\mathbf{A}$, датог са:

$$\mathbf{A} = \begin{bmatrix} \rho\cos\varphi & -\rho\sin\varphi & 0 \\ \rho\sin\varphi & \rho\cos\varphi & 0 \\ 0 & 0 & s \end{bmatrix} \tag{38}$$

закључујемо, да би услов (37) био задовољен потребно је да важи:

$$|s| < 1 \wedge |\rho| < 1. \tag{39}$$

Како вршимо пресликавање већег дела слике у мању, задовољили смо услов да је $|\rho| < 1$, али треба имати на уму и да мора бити задовољен и други услов тј. да је контраст мањи од 1.

На основу положаја и орјентације делова слике који се упоређују, одређују се ρ, φ, e и f док се s и o одређују тако да се минимизује растојање између делова слика који се упоређују. Како се врши оптимизација контраста и осветљења биће речено у наставку.

Закључимо, афине трансформације су погодне за употребу у фракталној компресији јер врше линеарно пресликавање, што даје једноставне изразе за израчунавање оптималних вредности контраста и осветљења. Да би афине трансформације биле контракција и тиме могле користити у фракталној компресији мора бити задовољен услов (39).

3.5. Оптимални коефицијенти афиних трансформација

Рекли смо, ако желимо да пронађемо оптималну трансформацију W, чији ће атрактор бити слика довољно блиска оригиналној, потребно је наћи, по колажној теореми, скуп трансформација w_i који пресликавају довољно близу већи део слике у мањи. Задатак се своди на израчунавање коефицијената s и o за које је минимизирано растојање између оригиналних делова слике и делова добијених пресликавањем трансформацијом w_i.

Када се већи део слике пресликава у мањи јавља се проблем да је потребно вршити упоређивање једног пиксела мањег дела слике са више пиксела већег дела слике. Из овог разлога потребно је пре израчунавања оптималних афиних трансформација извршити усредњавање већег дела слика у циљу добијања слика истих димензија. За добијање усредњене слике дупло мање величине потребно је да вредност пиксел ново формиране слике буде једнака средњој вредности четири пиксела веће слике.

Сл. 3.2. *Процес добијања оптималне вредности пиксела већег дела слике*

На Сл. 3.2. је приказан процес добијања вредности за пиксел веће слике који је најмањег растојања од одговарајућег пиксела мањег дела слике. Са слике се види да би смо добили пиксел најмањег растојања потребно је израчунати оптималне вредности за коефицијент контраста s и осветљења o.

Нека је $A(i)$ низ пиксела оригиналног дела слике, а $B(i)$ низ пиксела усредњеног већег дела дела слике који се пресликава у мању. За слику са n пиксела, Еуклитско растојање између датих делова слике је:

$$d_E = \sum_{i=1}^{n} \left[A(i) - s \cdot B(i) - o \right]^2 \tag{40}$$

Растојање d_E је функција од промељивих s и o, које постижу екстрем за вредности које задовољавају услов:

$$\begin{aligned} \frac{\partial d_E}{\partial s} &= 0 \\ \frac{\partial d_E}{\partial o} &= 0 \ . \end{aligned} \tag{41}$$

Из услова (41) одређују се оптималне вредности коефицијената s и o за које је растојање d_E минимално:

$$s_{opt} = \frac{n \cdot \sum_{i=1}^{n} [A(i)B(i)] - \sum_{i=1}^{n} A(i) \sum_{i=1}^{n} B(i)}{n \cdot \sum_{i=1}^{n} B(i) - \left[\sum_{i=1}^{n} B(i) \right]^2}$$
$$o_{opt} = \frac{\sum_{i=1}^{n} A(i) \sum_{i=1}^{n} B(i)^2 - \sum_{i=1}^{n} B(i) \cdot \sum_{i=1}^{n} [A(i)B(i)]}{n \cdot \sum_{i=1}^{n} B(i)^2 - \left[\sum_{i=1}^{n} B(i) \right]^2} \quad . \tag{42}$$

Одређивањем оптималних коефицијената s_{opt} и o_{opt} добија се трансформација која са најмањим Еуклидским растојањем пресликава већи део усредњене слике у мањи.

4. КОМПРЕСИЈА СЛИКЕ

На предходним странама бавили смо се само теориским делом фракталне компресије. У наставку следе алгоритами за реализацију фракталне компресије слике.

Основна идеја алгоритма за фракталну компресију слике је експлатација сличности већих и мањих делова слике, односно њених фракталних својстава. Као референтна слика у раду ће се користити слика Lenne[17] која је већ годинама незванични показатељ за оцену квалитета компресије. На Сл. 4.1. су приказане две слике Lenne. На десној слици су издвојена по два пара квадратних делова слике који су међусобно јако слични. Као што се може видети уз примену адекватног осветљења и контраста део шешира јако је сличан одразу већег дела шешира у огледалу. Исти је случај и са раменом.

Сл. 4.1. *Слика Lenna и слика са издвојеним сличним деловима слике*

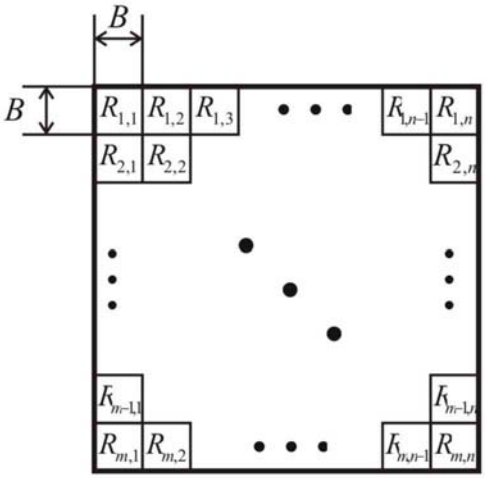
Ово су само два примера, а на целој слици се може уочити још много сличних примера. Основна идеја и задатак фракталне компресије управо и лежи у проналажењу сличних делова слике, као и коефицијената трансформације који ће пресликати веће делове слике у мање уз најмање растојање у усвојеној метрици.

Постоје више алгоритама за проналаженје сличних делова слике [5, 10, 11]. Ми ћемо кренути од најпростијег, али ће он дати најбољу (уводну) представу о алгоритму за фракталну компресију.

Алгоритам за фракталну компресију слике се може поделити у пет основна корака. У првом кораку слика која се жели компресовати се дели на мање делове, у аглосанксонској литератури они се називају ***ranges***, а у овом раду биће коришћен израз регион. Региони су блокови димензија $B \times B$ пиксела. Уобичајено је да је слика димензија, по основама, неки степен двојке, нпр. $2^8 = 256$, а узима се и да је B целобројни делилац броја пиксела по основи.

[17] Lenna Soderberg је швајцарскиња чија се слика налази у новембарском броју Playboy-а из 1972. Како је ова слика постала стандар на којој се одређује квалитет дигиталне обраде, као и оригинална слика може се наћи на web адреси http://www.cs.cmu.edu/~chuck/lennapg/

Слика се овом поделом претвара у дводимензину матрицу блокова пиксела, што је приказано на Сл. 4.2.



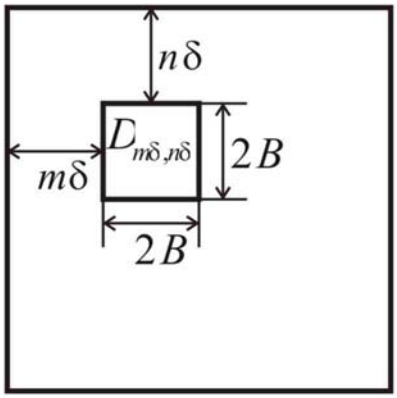
Сл. 4.2. *Подела слике на регионе*

Процес поделе слике се назива *сегментација* и у овом случају извршена је сегментација на једнаке квадратне блокове фиксне величине. Због оваквог начина сегментације слике овај алгоритам за фракталну компресију слике и спада у најпростији. Величином блока B одређен је степен компресије и квалитет слике. Што је већи блок, компресија је већа али се губи на квалитету слике, јер долази до изражаја блоковска структура слике. Слика се дакле сада представља низом блокова $\{R_i\}$, при чему је сваки блок описан координатом горњег левог угла (m, n) и величином странице блока B_i.

Да би се извршила компресија слике потребно је наћи трансформације w_i које пресликавају веће делове слика у регионе $\{R_i\}$. Када знамо положај и трансформацију којим се већи делови слике пресликавају у њима одговарајући регион, тада је слика у потпуности описана датим координатама и трансформацијом. Уколико се блок добијен од већег дела слике трансфромацијом w_i прихватљиво мало разликује од оригиналног блока тј. региона, тада ће се према колажној теореми и атрактор, добијен низом трансформацијом w_i, прихватљиво мало разликовати од датог региона.

Ако је количина информација о положајима и потребним трансформацијама мања од количине података потребних за опис слике, преношењем информације о сваком појединачном пикселу, извршена је компресија слике.

Делови слике који се пресликавају у регионе, морају бити већи од самих региона, како би био задовољен услов контрације. Већи делови слике се у аглосанксонској литератури називају ***domains***, а у овом раду бити коришћена одомаћена реч домен. Уобичајено је да домени буду већи од региона за 2^n број пута. Најчешћи је случај да су домени дупло већи, што је експрементално показано као најбоље решење. Скуп свих домена представља *простор претраге* Ω.

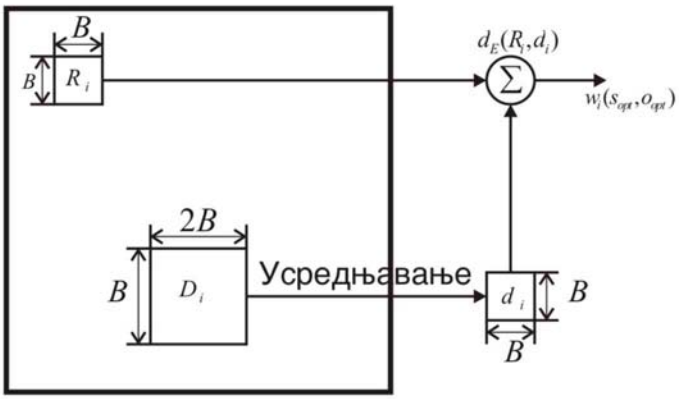


Сл. 4.3. *Формирање простора претраге* Ω

Домени су квадратни делови слике димензија, рецимо, $2B \times 2B$, а у потпуности су описани величином странице и положајем свог горњег левог угла чије су координате $(m\delta, n\delta)$, где је δ природан број који се може произвољно бирати. Што је δ мањи то је простор претраге већи, чиме се повећава могућност за проналажење сличнијег дела слике. Тиме се квалитет слике повећава, али се тиме и повећава време претраге као и потребна меморија за компресију што ће бити објашњено у наставку.

Други корак фрактална компресије слике, према горе реченом је формирање простора претраге Ω. Када је формиран простор и извешена сегментација слике прелази се на трећу корак компресије слике која представља суштину фракталне компресије.

У трећем кораку фракталне компресије врши се проналажење оптималних коефицијената трансформације w_i, и то на начин приказан на Сл. 4.4.



Сл. 4.4. *Поступак изтачунавања афиних трансформација*

Сваки појединачни регион R_i упоређује се са свим доменима D_i који чине простор претраге. Да би се упоредили региони са доменима, најпре се мора извршити усредњавање домена (на слици означен са d_i). Над тако усредњеним доменом може се приступити израчунавању Еуклидског растојања од региона, $d_E(R_i, d_i)$. Затим се према формули (42) рачунају оптимални коефицијенти афине трансформације $w_i(s_{opt}, o_{opt})$. Такође, памти се и

информација о величини и положају региона и домена. Када се регион упореди са свим доменима из простора претраге, памте се само информације о оном домену чије је Еуклидско растојање било најмање. Тиме смо добили транстормацију w_i^{best} чији ће атрактор бити најприближи региону R_i.

Постоје две методе у тражењу оптималног домена. Први метод је већ изложен у предходном пасусу и каже да се претрага за оптималним доменом врши кроз цео простор претраге, а затим се одлучује за онај домен који има најмање Еуклидско растојање од региона. Други метод предвиђа постојање прага толеранције. У овој методи претрага се врши кроз простор претраге све док се не наиђе на домен чије је Еуклидско растојање од региона мање од задатог прага толеранције. Мењањем прага толеранције утиче се на квалитет слике који је свакако лошији него у предходној методи, али се тиме такође смањује и време претраге, односно *смањује се време компресије слике* које представља једну од главних мана фракталне компресије слике.

За сваки регион слике R_i пронађена је трансформација w_i, одакле следи према колажној теореми да се атрактор дате транформације w_i прихватљиво мало разликује од региона R_i. Односно, пронађен је скуп трансформација тј. контрација $\{w_i\} = W'$, у односу на коју је слика *приближно инваријантна*. Атрактор контрације W' је приближно једнак оригиналној слици, па је слика која се компресује описана трасформацијама $W = \bigcup_{i=1}^{n} w_i$, чиме је извршена *фрактална компресија слике*.

Четврти и пети корак фракталне компресије слике су стандардни кораци у компресији. У четвртом кораку врши се *квантизација коефицијената трансформације*. Квантизација је одбацивање непотребних цифара из података. Односно компресија континуалних вредности информације на дискретне вредност тзв. квантизационе нивое. Број квантизационих нивоа мора бити одабран тако да покрива целу скалу вредности података како се не би изгубило на квалитету слике.

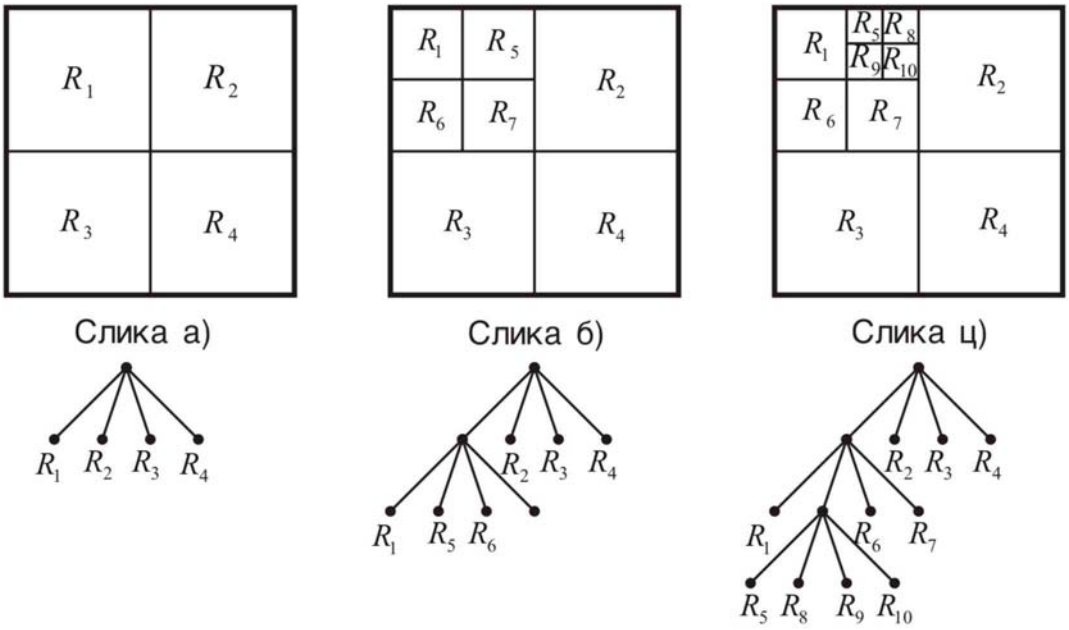
Пети корак у компресији слике није неопходна али је пожељна и представља *компресију коефицијената* транформације алгоритмом који не уноси губитке, на пример Хофмановим или LZW. У овом раду није коришћен ни један алгоритам за компресију коефицијената транформације, јер они представљају проблем и целину за себе, а нису директно повезани са суштином фракталне компресије.

Неопходне информације за декомпресију слике се смештају у излазни фајл и овим је завршена фрактална компресија слике.

4.1. Quadtree алгоритам

Применом основног алгоритма за фракталну компресију слике се може постићи задовољавајући степен компресије, али он пати од једног недостатка који је типичан и за JPEG формат компресије. Наиме, слика се дели на једнаке делове, односно на једнаке регионе. Међутим, слике нису униформне, неки делови слике садрже већу количину информација, односно детаља, од других. Овде ћемо изложити ***quadtree*** алгоритам [10, 11, 12, 13], који је коришћен у овом раду. Основна идеја quadtree алгоритма је експлатисање униформности делова слике. Слика се дели на блокове различитих величина према количини детаља који тај део слике садржи. Делови слике који садрже мање детаља се деле на веће блокове, а делови слике које садрже већу количину информација на мање блокове. *На овај начин може се постићи већи степен компресије и повећати квалитет слике*. Из предходно реченог намеће се закључак: применом quadtree алгоритма степен компресије се не може унапред знати, јер је он условљен самим садржајем слике.

У quadtree алгоритму као почетни члан простора претраге Ω узима се домен који преставља цела слика, док се за почетне чланове низа региона узимају четвртине слике. Затим се упоређује први регион у низу са доменом. Уколико је Еуклидско растојање домена и региона мање од задатог *прага толеранције*, памти се трансформација w_i и прелази се на следећи регион у низу. Ако Еуклидско растојање није мање од задатог прага толеранције регион и сви домени се деле на четири мања и израчунавање растојања се наставља са мањим регионом. Регион се упоређује са свим доменима који чине његов простор претраге, притом се памти трансформација w_i само за домен са минималним растојањем од региона. Поступак поделе региона на четири мања се понавља уколико је минимално пронађено растојање веће од задатог прага толеранције. При свакој подели региона на четири мања деле се и домени на четири мања који заједно са предходним доменима сада чине простор претраге за дати регион. На овај начин и за најмањи регион постоји двоструко већи домен. Quadtree алгоритам приказан је на Сл. 4.5.



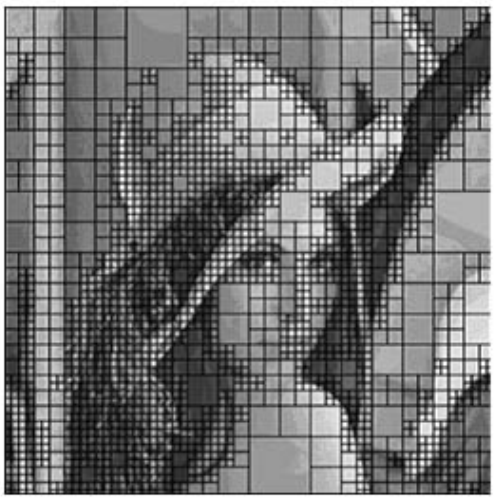
Сл. 4.5. *Графички приказ quadtree алгоритма*

На Сл. 4.5. приказана су три корака quadtree алгоритма. Прво се регион R_1 упоређивао са целом сликом, и установљено је да је растојање веће од задатог прага толеранције. Због тога је регион R_1 подељен на четири нова региона, а простору претраге су додати нови домени настали поделом сваког од постојећих домена (засад је то само један) на четири нова домена. Затим је упоређивањем установљено да постоји домен који је довољно близак новоозначеном региону R_1 па су у излазни фајл уписани одговарајући коефицијенти трансформације w_i. И на крају се прешло на следећи регион који се налази на истој *дубини гранања* као обрађени регион R_1, на Сл. 4.5. б) означен је као R_5. Видимо да регион 5 није имао задовољавајућу сличност ни са једним доменом из простора претраге зато се морао делити на мање регионе, наравно и домени су се делили на мање, итд. Испод слике са распоредом региона дата је презентација региона у облику стабла. Из овакве презентације се јасно уочава рекурзивна природа quadtree алгоритма, по којој је и добио име, јер се сваки чвор стабла (*tree*), ако се грана, грана на четири (*quad*) нове гране.

Одговарајући праг толерације се може одредити експериментално. Смањивањем прага повећава се осетљивост алгорима компресије на детаље и тиме добијена компресована слика мање одступа од оригинала, али смањивањем прага број региона ће се повећати, а самим тим и количина информација потребних за опис слике, што директно доводи до смањења нивоа компресије и повећање времена компресије. При повећању прага дешава се супротно. Зато праг толеранције треба да буде постављен тако да направи компромис између степена изобличења и нивоа компресије.

Упоређивање региона и домена није уобичајено да се врши још са четвртинама слике, јер је мала вероватноћа да ће растојање међу њима бити мање од задатог прага толеранције. Наиме, најчешће се са упоређивањем почиње тек када се региони сведу на блокове величине 32x32 пиксела, а до тада се само врши партиционисање слике. Такође, не треба дозволити ни

превелико усредњавање региона, па је потребно ограничити минималну величину региона. Ограничење величине региона са доње стране је потребно из саме чињенице да се жели компресовати слика, у супротном би смо могли доћи у ситуацију да нам је поребно већа количина информација за опис регион, него за пренос вредности његових пиксела. Минимални региони не би требали бити мањи од 4x4 пиксела. На Сл. 4.6. приказана је сегментација Ленине слике quadtree алгоритмом. На слици се може видети да се делови слике који садрже више детаља деле на ситније блокове, регионе.



Сл. 4.6. *Сегментација слике Lenna*

Упоређивање региона и домена је исто као и код основног алгоритма и описано је у предходној глави у склопу трећег корака. При томе треба водити рачуна да се региони упоређују само са доменима који су већи од њих, због услова контрактивности. Број домена са којим се упоређује регион из d-те дубине, ако се упоређивање врши на свим дубинама, дат је формулом:

$$D = \frac{4^d - 1}{3}. \tag{43}$$

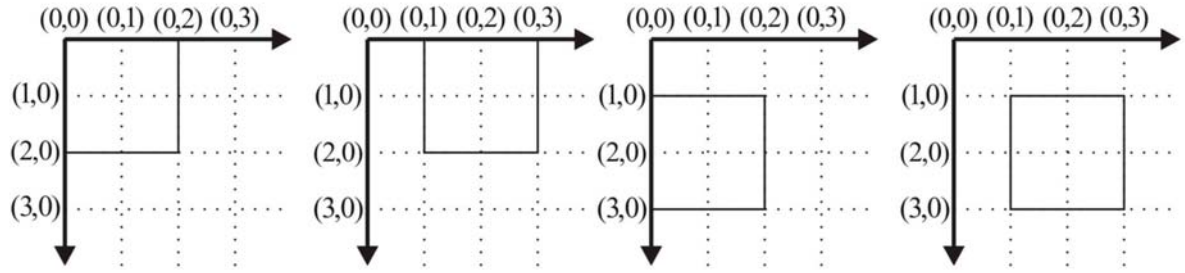
4.2. Формирање простора претраге

Најчешће се, а то је случај и у коду овог рада, простор претраге формира пре сегментације слике на регионе. Дакле, прво се формирају сви домени који чине простор претраге на свим дубинама, а касније, у процесу упоређивања, региони се пореде са већ формираном листом домена на жељеној дубину. На овај начин убрзава се компресија, јер није потребно при сваком повећању дубине изнова формирати листу домена за дату дубини. Наравно, на овај начин се повећава и меморијски простор потребан за фракталну компресију, јер је цео простор претраге за све време компресије смештен у меморији.

Због рекурзивне природе алгоритна и великог броја домена време извршавања компресије може трајати неприхватљиво дуго. Управо у овој чињеници лежи и главна мана фракталне компресије, што ће касније бити мало детаљније анализирно. У циљу скраћивања времена извршавања програма развијани су алгоритми који ефикасно редукују простор претраге. Једна од

основних техника је да се претражују домени који су најмање већи од региона, јер је вероватноћа налажења домена довољно блиског региону далеко већа у овом делу простора претраге. У овом раду, због једноставности, региони се упоређују само са доменима који су дупло већи од њих.

Да би се домени могли упоредити са регионима они се морају усреднити, како је то објашњено у глави 3.5. Стога није неопходно памтити домене у њиховим оригиналним величинам, већ само њихове усредњене вредности, на овај начин је уједно и смањен меморијски простор потребан за компресију. Како се региони упоређују само са доменима дупло веће величине, усредњавање се може извршити пре формирања простора претраге. Од оригиналне слике могу се формирати четири усредњене слике у зависности од пиксела са којим се почиње са усредњавањем. Почетни пиксели могу бити (0,0), (0,1), (1,0) или (1,1), као што је приказано на Сл. 4.7.



Сл. 4.7. *Формирање четири различите успредњене слике*

Ивични пиксели који недостају приликом усредњавања најчешће се или попуњавају средњом вредности преосталих пиксела или нулама. Из свега горе реченог види се да формула (43) за број домена који чине простор претраге најчешће не одговара пракси па је погодније користити формулу:

$$D = 4 \cdot \sum_{d=d_{\min}}^{d=d_{\max}} \left(\frac{N}{n_d} \right)^2 . \qquad (44)$$

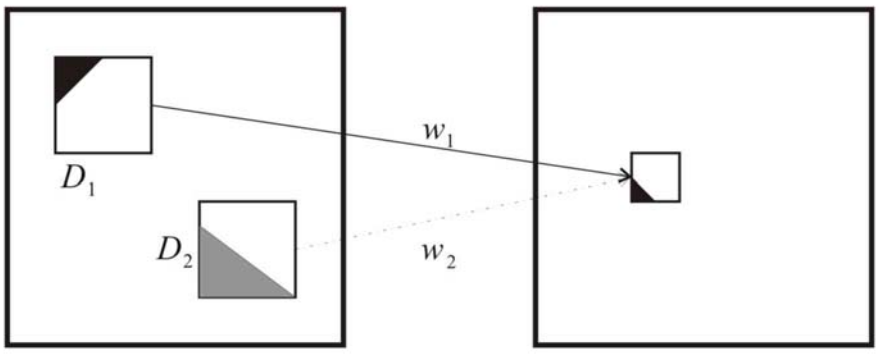
У формули (44), D је укупан број домена, d дубина, N број пиксела по једној ивици (предпостављено је да је слика квадратног облика), n_d величина стране домена на d-тој дубини а $d_{\max}$ и $d_{\min}$ су максимална и минимална дубина.

У овом раду нису коришћене четири усредњене слике већ само једна.

Тек након овако формираног простора претраге прелази се на партиционисање слике на регионе рекурзивним quadtree алгоритмом. У току партиционисања врши се проналажење оптималне трансформације за дати регион и на преостала два корака компресије слике које су исте као и код основног алгоритма, а то су: квантизација и компресија коефицијената трансформације.

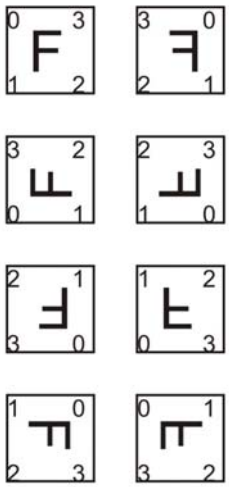
4.3. Класификација домена и региона

Из свега досад реченог може се закључити да суштина фракталне компресије лежи у међусобном упоређивању делова слика. Потребно је пронаћи веће делове слике (домене) који имају најмање растојање од делова добијених quadtree партиционисањем слике (региони). Временски овај део компресије представља њен најкритичнији део. Из овог разлога формирање целокупног простора претаге се извршава пре почетка упоређивања, такође ће се, у циљу убрзања компресије, региони упоређивати само са доменима дупло веће величине. Али и овако редукован простор претраге представља огромну количину могућности за проналажење домена најмањег растојања. Погледајмо за тренутак Сл. 4.8.



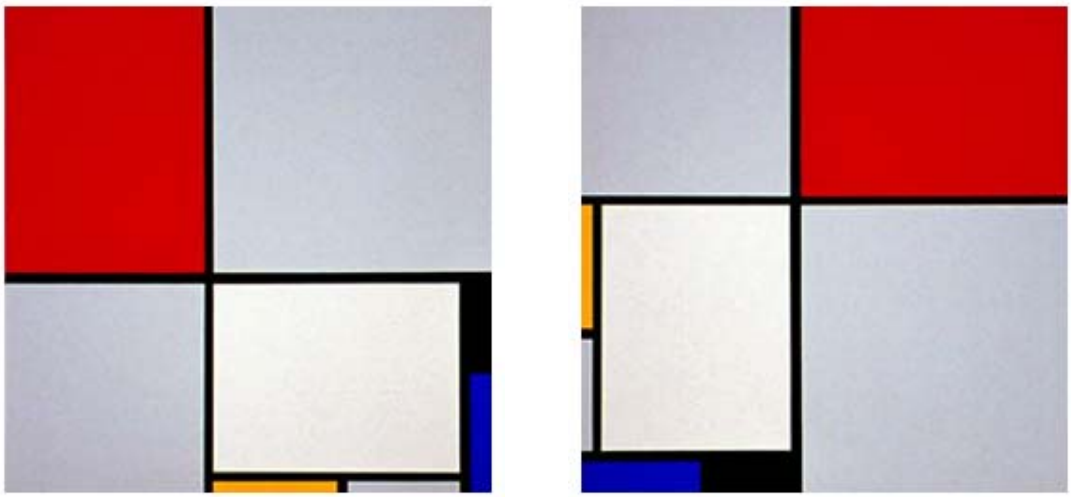
Сл. 4.8. *Пресликавање са и без могућности ротације*

На Сл. 4.8. се види да је домен 1 ближи региону него домен 2, али домен 1 је ротиран у односу на регион за 90°. Из овога се може закључити да, у колико се жели пронаћи домен најближи региону, региони се морају упоредити са свим потенцијалним ротацијама домена. Слику, односно домен, је могуће ротирати на 4 начина, а могуће је окретати слику као у огледалу, тако да је укупан број *орјентација слике* 8. Тиме би се простор претраге увећао осам пута. Орјентацију слике је најлакше приказати ако би се сваки угао слике нумерисао бројем од 0 до 3. На Сл. 4.9. приказани су могуће орјентације исте слике.



Сл. 4.9. *Орјентације слике*

Дакле уколико би се делови слике који се упоређују довели у *каноничан положај*, један у односу на други, не би било потребно упоређивати све орјентације домена са регионом. Шта би био тај каноничан положај и како уопште да знамо у ком се положају налази слика? На Сл. 4.10. нама је интуитивно јасно да је лева слика у горњем реду каноничан положај за преосталих седам, јер смо на њој препознали латинично слово **F**. Међутим, шта радити са сликама или деловима слика када нам је садржај непознат? Погледајмо једну слику орјентисану на два начина.



Сл. 4.10. *Која је орјентација канонична?*

Познаваоци апстрактне уметности би могли са сигурношћу да нама кажу који је од ова два положаја ″каноничан″, али нама није ни циљ да препознамо слику и њен садржај, већ само да је на неки рачунарски прихватљив начин опишемо.

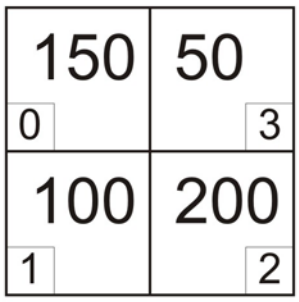
4.3.1. Класификација на основу интензитета осветљења

У светлу горе предложене нумерације делова слике, можемо узети да је, рецимо, каноничан положај слике онај положај у ком се најсветлија квадратна четвртина слике налази у горњем левом углу. На овај начин ми слику или њене делове посматрамо, односно описујемо, као средњу вредности пиксела њених квадратних четвртина. Како слика сада представља пермутацију осветљења њена четири квадранта може се закључити да постоји 4!=24 различитих нумеричких изгледа слике.

Дакле, све слике се међусобно могу разврстати у 24 различитих класа према распореду средњих вредности пиксела који чине квадратну четвртину слике. Али, као што смо горе већ напоменули једна слика може да се орјентише на 8 начина. Ово имплицитно значи да, дељењем слике на четири квадранта, можемо издвојити 3 различите врсте слика где свака врста тј. група има по 8 орјентација. На Сл. 4.11. је приказан канонични положај слике из сваке групе.

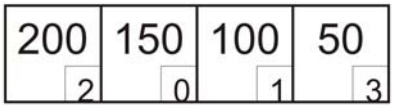
Сл. 4.11. *Каноничани положаји слике*

Ради прегледности и разумевања кода, изложићемо у табели све могуће ″нумеричке″ изгледе региона, односно домена, као и класе којима припадају и њихове орјентације. Али најпре морамо објаснити нумерацију која је коришћена у програму, а коју сада желимо користити у табеларном приказу изгледа произвољног блока слике. Усвојимо исту нумерацију квадраната блока као и приликом објашњења орјентације слике, означићемо горњи леви квадрант бројем 0, затим доњи леви бројем 1, па доњи десни бројем 2 и на крају горњи десни бројем 3. Нумерација квадраната блока на Сл. 4.12. је приказна мањим бројем у углу квадранта.



Сл. 4.12. *Блоковски изглед произвољне слике*

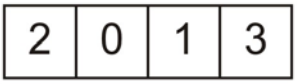
Погледајмо сада један од могућих распореда средњих вредности пиксела квадратних четвртина блока произвољне величине датих већим бројем на средини квадранта (Сл. 4.12.). Закључујемо да је најсветлији квадрант доњи десни јер је средња вредност његових пиксела 200. Најтамнији квадрант је горњи десни јер је средња вредност његових пиксела 50. Ако сада сложимо у низ, од најсветлијег квадранта ка најтамнијем, податке о средњим вредностима пиксела квадраната заједно са податком у ком квадранту су се налазили, Сл. 4.12. добија нумерички изглед:



Сл. 4.13. *Средње вредности квадраната слике сложених у низ по нерастућем редоследу*

За класификацију слике према интензитету светлости, нас више не занима стварна вредност појединих квадраната, већ само њихов редослед. Дакле, за класификацију и нумерички запис слике потребано је познавати редослед нумерисаних квадраната, који су на Сл. 4.13. дати у доњем десном

углу сваког квадрата. За произвољан распоред интензитета осветљења дат на Сл. 4.12. имаће се нумерички запис приказан на Сл. 4.14.



Сл. 4.14. *Нумерички запис изабране произвољне слике*

Потребно је разумети да ће све слике који имају исти распоред интензитета светлости, тј. најсветлији квадрант је доњи десни, па горњи леви, затим доњи леви и најтамнији горњи десни, имати исти нумерички запис као овај дат на Сл. 4.14.

Значи, да би смо добили нумерички изглед домена или региона, квадранте блока слике смо нумерисали на једнозначан начин, а затим смо бројеве квадраната сврстали у низ тако да се на првом месту нашао број квадранта у коме је интензитет осветљења највећи, а на последњем месту број квадранта коме је интензитет осветљења најмањи.

Могло се користити и простије означавање али због даљег разумевања, а нарочито разумевања кода коришћеног у овом раду било је неопходно овладати оваквом нумерацијом блока слике.

Сада можемо изложити табелу 1. у којој се налазе сви нумерички изгледи блокова слика, региона или домена, произвољне величине, као и класе којима припадају односно њихова орјентација, на основу распореда интензитета осветљења њихових квадратних четвртина.

Класа	Орјентација	Нум. запис	Класа	Орјентација	Нум. запис	Класа	Орјентација	Нум. запис
0	0	0213	1	0	0123	2	0	0132
0	1	1320	1	1	1230	2	1	1203
0	2	2031	1	2	2301	2	2	2310
0	3	3102	1	3	3012	2	3	3021
0	4	0231	1	4	0321	2	4	0312
0	5	1302	1	5	1032	2	5	1023
0	6	2013	1	6	2103	2	6	2130
0	7	3120	1	7	3210	2	7	3201

Табела 1. Нумерички изгледи слике према нивоу осветљења квадраната

Изложимо сада математичко тумачење овакве нумерације блокова слика у циљу сагледавања њених предности.

Лако се уочава да је прва цифра у нумеричком запису иста са орјентацијом за прва четири члана сваке класе. Ово смо и желели да постигнемо јер смо практично првом цифром рекли колико је најсветлији квадрант удаљен од горњег левог угла слике крећући се од тог угла супротно казаљци на сату. Односно, прва цифра нам говори колико пута треба слику да ротирамо за $\frac{\pi}{2}$ тако да се она нађе у каноничном положају. Када је орјентација већа од 3 значи да морамо најпре извршити огледалско пресликавање слике па тек онда вршити ротирање.

Погледајмо сада нумеричке записе слика али само каноничан положај сваке од три класа. Примећује се да је број класе једнак броју цифара између броја 0 и броја 2! Када се од сваке цифре нумеричког записа одузме прва цифра (за коју смо видели да представља орјентацију) по модулу 4 добиће се:

- за прве четири орјентације, каноничан положај класе којој је нумерички запис припадао;
- за друге четири орјентације цифре 1 и 3 ће заменити места, а положај бројева 0 и 2 ће остаће исти, јер смо рекли да дефинише којој класи блок слике припада

Примећујемо да уколико се цифра 1 налази испред цифре 3, да би се блок довео у каноничан положај, није потребно вршити огледалско пресликавање.

Резимирајмо речено у виду табеле 2.

Класа	Каноничан положај и њена ротација	Огледалско пресликавање
0	**02**13	не
	0231	да
1	**0**1**2**3	не
	03**2**1	да
2	**0**13**2**	не
	031**2**	да

Табела 2. Дефиниција класе према каноничном положају слике

Други ред у табели 2. добијен је тако што је од сваког нумеричког изгледа блока слике одузета прва цифра у низу по модулу 4. Од растојања 0 и 2 се може закључити којој класи блок слике записан на нумерички начин припада, а према распореду цифара 1 и 3 да ли је потребно извршити огледалско пресликавање да би се блок довео у каноничан положај.

Закључимо, приликом формирања простора претраге домени су сврстани у три класе. У току извршавања quadtree алгоритма за сваки регион ће се такође израчунати којој класи припада и регион ће се упоређивати само са оним доменима који припадају истој класи. Да би се извршило упоређивање потребно је знати да ли су домен или регион огледалски пресликани и колико су ротација за $\frac{\pi}{2}$ удаљени од каноничног положаја.

4.3.2. Класификација на основу варијансе

Класификацијом слика у три класе и према орјентацији, време фракталне компресије је сведено на подношљивих пар секунди, али је то и даље јако дугачак временски период. Зато метод који су предложили Фишер[18] [12, 13, 14], Јакобс[19] и Бос[20] подразумева поред класификације слике према нивоу осветљења, такође и класификацију према варијанси квадрантних четвртина слике. Овај метод спада у један од последњих новина у пољу фракталне компресије слике.

Поред рачунања средње вредности пиксела квадратне четвртине делова слике, могуће је израчунати и варијансу интензитета пиксела те четвртине. Која информација се може извући из податка о варијанси? Варијанса је одступање променљиве од његове средње вредности (очекиване вредности) и дефинише се формулом:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{N}\left(z_i - \bar{z}\right)^2 \tag{45}$$

где је n број пиксела квадратне четвртине блока (домена или региона), а $\bar{z}$ је средња вредност пиксела блока, дефинисана са:

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{N} z_i \tag{46}$$

Варијанса нам говори колико се блок разликује од своје средње вредности, односно колико на датом делу слике има детаља. Ако дати блок има наглих прелаза нивоа осветљења, какав је случај са ивицама, варијанса таквог блока ће бити већа, док за блокове који садржи уједначени интензитет осветљења, који практично не преносе скоро никакву информацију, варијанса ће бити мала.

У зависности од интензитета варијансе квадратних блокова, постоји 4!=24 могућих распореда четвртина. На основу тога домени и региони се могу сврстати у 24 класе. Ради прегледности табеларно биће изложене све могуће пермутације варијансе. Задржимо нумерацију коришћену у класификацији блокова слике на основу интензитета осветљења: бројем од 0 до 3 означени су квадранти унутар блокова. Формиран је низ од броја квадраната тако да је први елемент у низу број квадранта који садржи највећу варијансу, а последњи број у низу је број квадранта у коме је варијанса најмања. У табели 3 изложени су могући распореди варијанси по квадрантима према предходно објашњеној нумерацији, као и класе којима такав распоред варијанси припада.

[18] Yuval Fiser, аутор књиге ˝Fractal Image Compression: Theory and Application to Digital Image˘ Springer-Verlag Telos, 1995.
[19] Bill Jacobs
[20] Roger Boss

Број	Распоред варијансе	Класа	Број	Распоред варијансе	Класа
1	0123	0	13	2013	8
2	0132	1	14	2031	10
3	0213	2	15	2103	14
4	0231	4	16	2130	20
5	0312	3	17	2301	16
6	0321	5	18	2310	22
7	1023	6	19	3012	9
8	1032	7	20	3021	11
9	1203	12	21	3102	15
10	1230	18	22	3120	21
11	1302	13	23	3201	17
12	1320	19	24	3210	23

Табела 3. Нумерација слике према варијанси квадраната

У колико постоји класификација домена и региона према варијанси њихових квадраната, тада ће се региони поредити само са доменима који припадају истој класи.

4.3.3. Упоређивање региона са доменима

До сада смо се бавили само сужавањем простора претраге на оне домене за које, на основу неког од поменутих критеријума, предпостављамо да су потецијални кандидати за добијање најмањег растојања од региона. Објаснимо сада саму процедуру упоређивања региона и домена.

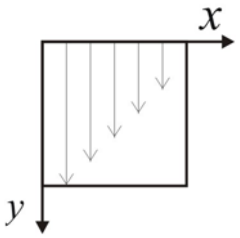
Циљ упоређивања је да се нађе домен који се са најмањим Еуклидским растојањем пресликава у регион. У том циљу смо за сваки домен из простора претраге одредили којој класи припада и које је орјентације. Постоје две класификације, прва класификација је према интензитету светлости, а друга је према интензитету варијансе. Сада се регион пореди само са доменима који припадају истој класи унутар прве и друге класификације. Како нам је позната орјентација и домена и региона, упоређивање се врши само дуж једне орјентације. Није економично, са становишта времена конверзије, окретати домене и регионе да би били постављени у њихов канонични положај, а такође то није ни неопходно радити. Познавањем орјентације домена и орјентације региона, може се закључити којим редоследом треба извршити упоређивање.

На овом месту би требало још једном нагласити да је упоређивање делова слике на основу њихове орјентације могуће само уколико они припадају истој групи унутар класификације на основу интензитета светлости.

Упоређивањем се израчунава минимално растојање између региона и одговарајућег домена како је то објашњено у глави 3.5. Да би смо дошли до жељене вредности за минимално растојање потребно је упоредити пикселе региона са пикселима домена који се налазе на истој позицији унутар блока слике. Како се и домени и региони могу наћи у било ком од 8 положаја (8 орјентација), закључује се да минимално постоји такође 8 различитих редоследа упоређивања њихових пиксела. Из тог разлога разумљиво је да

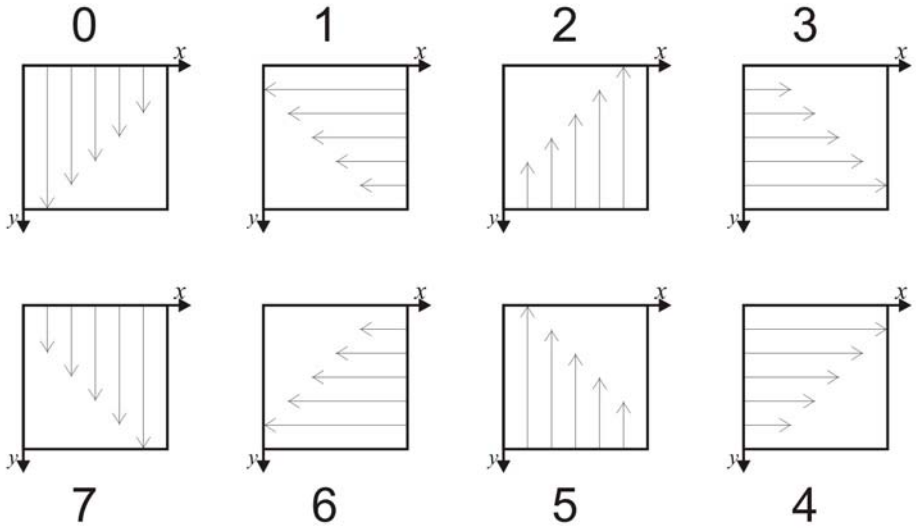
орјентација региона или домена неће бити иста са смером скенирања, јер би у том случју постојало 8x8=64 могућих начина скенирања.

Скенирање пиксела домена обављаће се увек истим редоследмо, најпре се скенирају колоне (y оса) па редови (x оса). На Сл. 4.15. графички је приказан редослед скенирања домена, где је са највећом стрелицом приказано шта се прво скенира, а са најмањом шта је последње.



Сл. 4.15. *Скенирање домена*

Како се домен увек скенира на јединствен начин, кроз регион, да би се извршило правилно упоређивање, скенирање се мора прилагодити орјентацији домена и самог региона. За скенирање кроз регион постоје зато 8 смерова, који су приказани и нумерисани на Сл. 4.16.



Сл. 4.16. *Смерови скенирање региона*

Који смер скенирања ће бити потребан за упоређивање региона и домена ако су познате њихове орјентације може се израчу према формулама:

$$Smer\_skeniranja = (4 + dom\_orj\%4 - rang\_orj\%4)\%4 \qquad (47)$$

или

$$Smer\_skeniranja = (4 + (dom\_orj\%4 + 3*(4 - rang\_orj\%4))\%4)\%8 \qquad (48)$$

Формула (48) се користи када је један домен или регион огледалски пресликан. У сличају да ни домен ни регион нису огледалски пресликани, или су оба огледалски пресликана, треба користити формулу (47).

Треба напоменути да је поређење могуће вршити и са негативном скалом контраста. Ако погледамо Сл. 4.17. може се приметити да је лева слика индентична десној, ако црна и бела боје замене места.

Сл. 4.17. *Индентичне слике при различитим склама осветљења*

У овом раду није експлатисана могућност поређења региона са доменима коришћењем негативног контраста. У колико би се желео користити и овај метод поређења потребно би било повећати број бита контраста за један да би се задржао исти квалитет компресоване слике.

Овим је покривена основна теорија потребна за фракталну компресију слике применом quadtree алгоритма са класификацијом домена на основу варијансе и интензитета осветљења. У наставку следи теорија потребна за разумевање декомпресовање овако компресоване слике.

5. ДЕКОМПРЕСИЈА СЛИКЕ

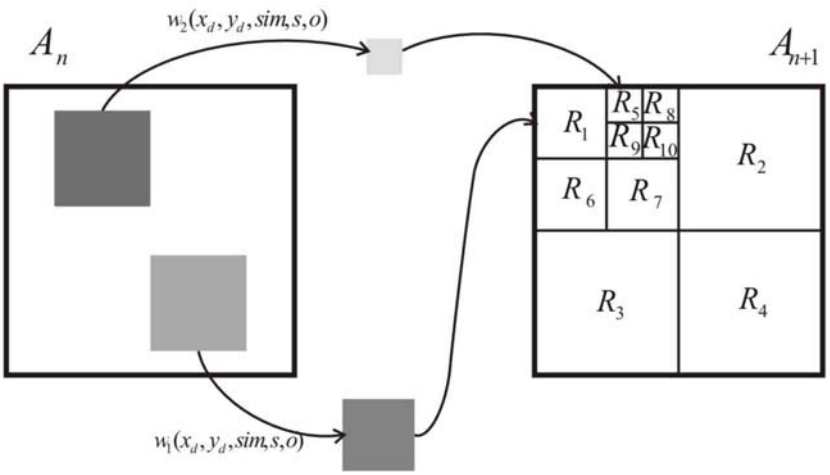
Досада смо се детаљније бавили само алгоритмом за компресовање слике. Да би смо компресовану слику, записану у излазном фајлу, могли превести у матрицу пиксела, односно слику, мора постојати инверзан програм програму за компресију. Декомпресија фрактално компресоване слике је, за разлику од компресије, веома једноставна, јер се у потпуности ослања на Банахов став, приказан у формули (10). Једном, када прочитамо коефицијенте трансформације, као што се то и из поменуте формуле јасно види, више нас не занима да ли је компресија извршена основним, quadtree или неким другим алгоритмом. Све што треба да урадимо је да за сваки регион применимо одговарајућу трансформацију, а затим цео поступак поновимо онолики број пута потребан да се добије жељена минимална разлика дата Колажном теоремом. Из приложеног се види да алгоритам за декомпресију слике осим што је једноставан он је и изузетно брз.

Декомпресија се обавља у три корака. У првом кораку се врши декомпресија коефицијената трансформације који су компресовани алгоритмом примењеним у петом кораку компресије слике, рецимо Хофмановим или LZW алгоритам. Као што смо напоменули, пети корак није неопходна за фракталну компресију и није коришћена у овом раду па је стога прескочен и први корак декомпресије.

Други корак у декомпресији претставља формирање естимираних коефицијената алфиних транформација од квантизованих вредности добијених у првом кораку.

У већини програма за декомпресију први и други корак се извршавају упоредо. Наиме, уобичајено је да се најпре све трансформације прочитају из улазног фајла и сместе у меморију, па да се тек онда пређе на формирање слике. На овај начин захтева се више меморије за декомпресију али је време извршавања краће.

Трећи корак је фаза формирања слике. Као што је то у уводном делу речено, за почетну слику узима се произвољна слика. Најлакше је, што је и у овом раду искоришћено, да се за полазну слику узме матрица, одговарајуће величине, са вредношћу чланова од неке средње вредности опсега пиксела. Регион нове слике R_i се формира тако што се на домен предходне слике примени афина трансформација w_i, чиме је већи део слике пресликан у мањи.



Сл. 5.1. *Формирање декомпресоване слике*

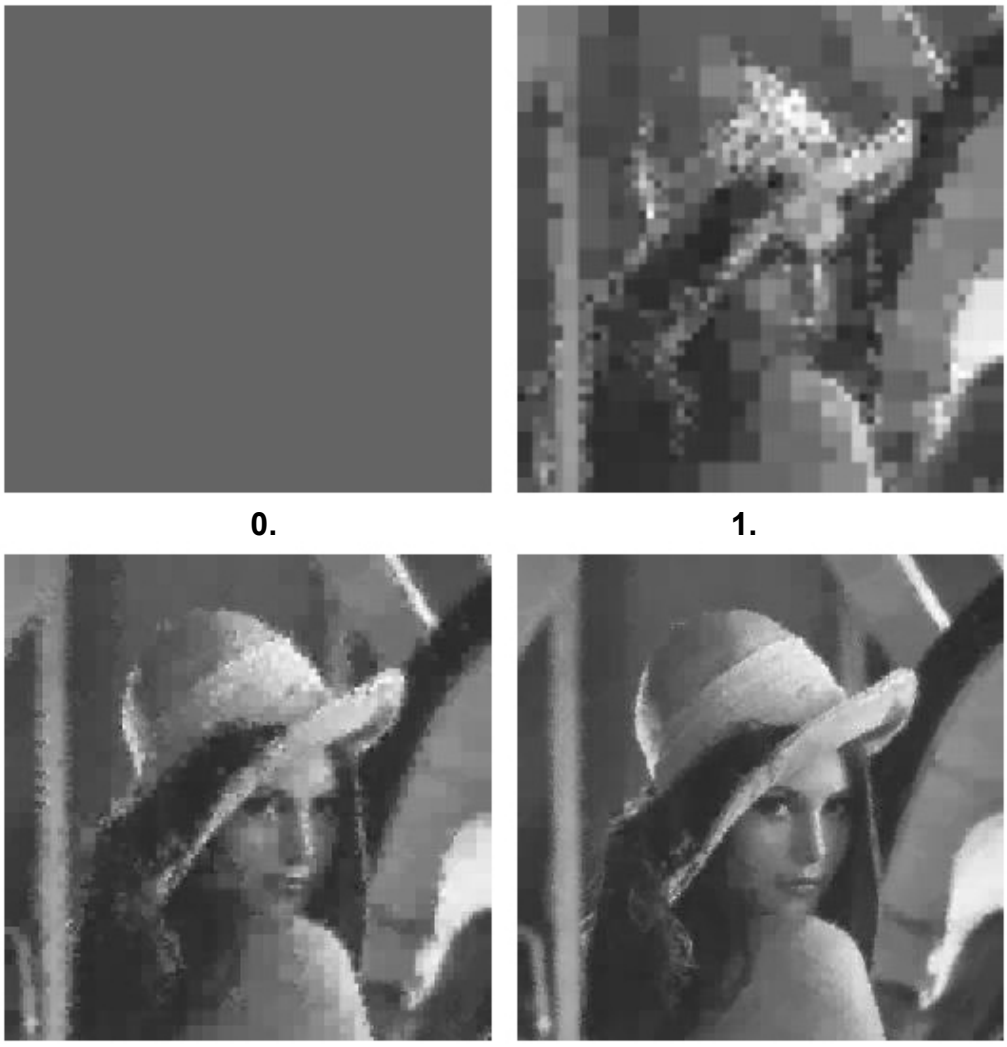
На Сл. 5.1. приказан је трећи корак декомпресије слике. Може се видети како се од слике A_n формира слика A_{n+1}. Блок R_1 слике A_{n+1} се формира тако што се на домен, који се у њега пресликава, примени афина трансформација $w_1(x_d, y_d, sym, s, o)$. Параметри унутар заграде трансформације w_1 означавају да се координате домена као и смер скенирања, заједно са контрастом и осветљењем, налазе унутар афиних трансформација добијених у процесу компресије. Слика A_{n+1} се формира тако што се на A_n примене све трансформације w_i чиме се формирају сви блокови R_i. Када се формира слика A_{n+1} почиње се са формирањем слике A_{n+2} тако што се опет све трансформације w_i сада примене на слику A_{n+1}. Процес се наставља све док се не постигне жељено минимално растојање:

$$d(A_N, W(A_N)) < \varepsilon \tag{49}$$

тада је A_N атрактор трансформације W. Овим је декомпресија завршена.

Број итерација N ретко прелази 10. Само после пар итерација јасно се може препознати садржај слике.

На Сл. 5.2. виде се прве три итерације декомпресије слике. За генерисање слике коришћен је код овог рада. За почетну слику је узета униформно сива слика. После прве итерације виде се контуре слике и може се препознати да је на слици лице неке особе. А већ после друге итерације јасно је да је реч о Лени. Уочавамо да је процес конвергентан и да је атрактор трансформације врло сличан оригиналној Лениној слици (Сл. 4.1.).



Сл. 5.2. *Почетна и прве три итерације декомпресије слике*

Предходно изложени алгоритам за декомпресију коришћен је у овом раду. Иако је веома брз, данас су развијени алгоритми са краћим временом извршавања и ефикаснијем искоришћењу меморије [11].

Фрактална компресија је блоковски заснована техника. Као свака таква техника пати од изражајног блоковског изгледа слике, нарочито при великом степену компресије. Како је фрактална компресија рекурзивне природе блоковски ефекат се додатно може повећати. Развијене су разне технике постпроцесирања слике [15, 16] које су засноване на *интерполацији*. На Сл. 5.3. десно дат је увећан део Ленине слике у чијем декомпресовању није примењено постпроцесирање, а лево од ње је исти тај део слике када се примени тзв. углачавање слике. На сликама је увећан контраст ради лакшег уочавања блокова.

Сл. 5.3. *Lenna са и без примењене интерполације*

У самом коду овог рада није коришћена опција за формирање декомпресоване слике различитих величина, што је једна од карактеристика ове компресије која је издваја од других. Међутим, извршена је анализа и са постојањем ове опције, што ће бити изложено у закључку. Ако се узме у обзир и могућност скалирања слике, код програма се само незнатно мења, у виду множења одговарајућих коефицијената трансформације скаларним фактором.

6. ПРОГРАМ

У овој глави биће објашњен садржај фајлова коришћених у раду, као и основне фунукције и поједини делови кода потребни за лакше разумевање програма.

Код рада ″Фрактална компресија слике″ написан је у програмском језику C [6], Microsoft Visual Studio 6, а графичко окружење програма са резултатима компресије урађено је у MATLAB-у 6.5 [7].

Програм за фракталну компресију слике није stand alone апликација, што значи да за његово покретање морамо поседовати MATLAB инсталиран на рачунару. Да би смо покренули програм потребно је да MATLAB-ов Current Directory буде *Compression* и да у Command Window упишемо *fractal*.

Фајлови програма који се налазе у Compression фолдеру су:

1. vfm.m
2. vfm.dll
3. fractal.fig
4. fractal.m
5. EncR.cpp
6. EncG.cpp
7. EncB.cpp
8. DecR.cpp
9. DecG.cpp
10. DecB.cpp

Објаснимо садржај свакога фајла.

6.1. Фајлови Vfm.m и Vfm.dll

У овом раду компресовали смо реалну слику добијену са web камере. Да би смо слику са камере уопште могли добити у MATLAB-у нужно је поседовати фајлове *vfm.dll* и *vfm.m*. Код се бесплатно може скинути са интернета [18]. Vfm (скраћено од Vision for MATLAB) омогућује да се преко Matlaba приступи драјверима видео уређаја као што су USB web камере. Vfm код се ослања на Microsoft's Video for Windows драјвер (драјвери web камере морају, наравно, бити инсталирани) и омогућава њихово подешавање. Такође омогућава да се једноставном командом из Matlaba дохватити једна (или више) слика са камере, а њихов садржај прикаже у одговарајућем формату. У Matlab-у слика се памти као матрица вредности пиксела.

У наставку следи проста наредба којом се једна ухваћена слика са USB камере смешта у matlab-ову променљиву I .

$$I = vfm\ (\text{'}grab\text{'}, 1);$$

6.2. Фајови fractal.fig и fractal.m

Фајл *fractal.fig* је графичко окружење програма и не захтева неко посебно објашњене.

Matlab-ов фајл који се покреће на неки догађај из графичког окружења је *fractal.m*. Притиском на дугме *Enkoduj* у GUI окружењу извршава се део кода фајла fractal.m који врши гребовање слике са камере и њено компресовање.

Да би се поједноставио код за компресију, слика која се компресује је фиксне величине 256x256 пиксела. Како слика са камере може бити различите величине, извршено је одсецање делова слике који се не компресују. Такође слике са камере је у RGB систему, како је код за компресију и декомпресију писан за gray слику, сваки канал (боја) се мора посебно компресовати и декомпресовати.

Да би фајлови за компресију, EncR.cpp, EncG.cpp и EncB.cpp, који су написани у програмском језику C, могли користити у Matlab-у потребно их је превести у MEX фајлове. MEX фајлови су извршни фајл који изиграва дењену библиотеку функција. На Windows оперативном систему они добијају екстензију .dll (**d**inamic **l**ink **l**ibrary). Једном преведен C фајл у MEX може се користити на свим платформама истог оперативног система. Покретање dll фајла, односно извршење компресије, врши се навођењем имена dll фајла у коду Matlab-а.

Не компресована слика се на крају приказује у првом прозору GUI окружења ради визуалног упоређивања компресоване слике.

Притиском на дугме *Dekoduj* у GUI окружењу извршава се део кода фајла fractal.m који врши декомпресију фрактално компресоване слике, израчунава степен компресије као и PSNR (**P**eak **S**ignal to **N**oise **R**adio).

Како је то био случај са фаловима за компресију и фајлови за декомпресију се морају најпре превести у dll фајлове. Сваки фајл компресоване боје се засебно декомпресују, а затим се формира RGB слика. Формирана слика се приказује у десном прозуру GUI окружења, након што се израчуна степен компресије и PSNR фактор.

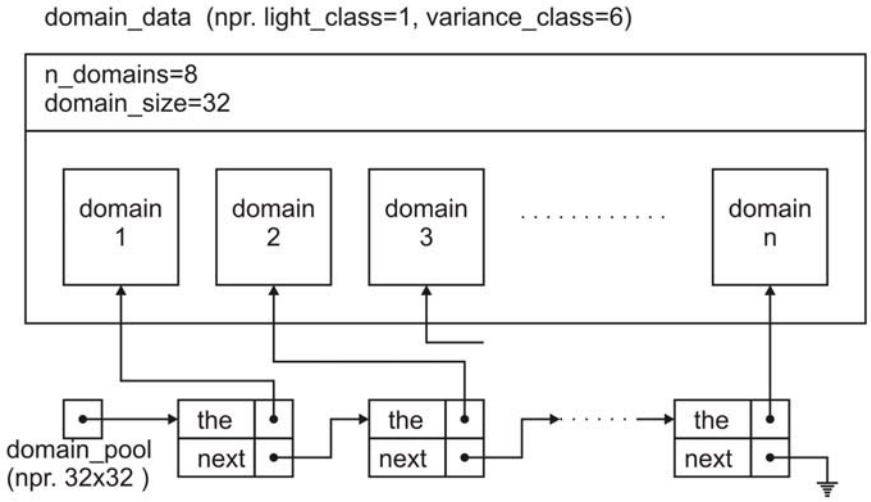
6.3. Фајлови EncR.cpp, EncG.cpp и EncB.cpp

Слика добијена са камере је у RGB формату, што значи да је сваки пиксел слике формиран од збира интензитета три основне боје, црвене (R), зелене (G) и плаве (B). Код за компресију слике има за улаз gray слику (слика у нивоима сиве), зато се свака боја морала засебно компресовати и декомпресовати. Наравно, свака боја понаособ представља gray слику за себе. Програми за компресију ће, стога, сваки направити свој излазни фајл, тако да се у излазном фајлу outR.frac налази компресована црвена компонента слике, у outG.frac зелена, а у outB.frac плава.

Довољно је проћи кроз код компресије једне од боја, јер је једина разлика између фајлова за компресију у називу излазног фајла.

6.3.1. Код за компресију

Ради бољег разумевања кода потребно је најпре објаснити структуре коришћене у коду. Као што је то раније речено, простор претраге формира се пре почетка quadtree алгоритма да би се обезбедила бржа и ефикаснија претрага домена. Простор претраге је представљен једноструко уланчаном листом домена који припадају истој структури. Величина домена може бити 32x32, 16x16 или 8x8 пиксела, на основу тога простор претраге се дели у три групе. У свакој од ове три групе домени су разврстани према класи којој припадају унутар класификације на основу интензитета осветљења и класификације на основу варијансе. Домени истих класа дефинисани су заједничком структуром коју чине подацима о њиховој величини и броју, као и подстуктуром за опис домена. Сваки домен појединачно дефинисан је својом позицијом, ротацијом, сумом и квадратном сумом пиксела.



Сл. 6.1. *Шематски приказ коришћене структуре*

На Сл. 6.1. је шематски приказана уланчана структура простора претраге домена који су нпр. величине 32x32, припадају првој класи према распореду осветљења и шестој класи према распореду варијансе. Напоменимо још једном да у коду програма овог рада није коришћена класификација на основу варијације, али су нека тестирања квалитета компресије вршена и са њом, па је због општости наведена.

Сада можемо прећи на основне фунције коришћене у коду за компресију слике. У коду се могу издвојити пет основних функција:

1. pool
2. classify
3. quadtree
4. compare
5. pack

Фунција *pool* алоцира меморију за податке о домену и израчунава их. Подаци о доменима се памте у једноструко уланчаним листима структура

према предходно објашњеном принципу. Да би се формирала структура потребно је да се изврши класификација сваког домена.

За класифинацију блока слике користи се функција *classify*. Функција classify има као повратну вредност суму и квадратну суму вредности пиксела блока слике чија се класификација врши. Такође она враћа податак о орјентацији блока као и броју класе којима блок припада.

Након формирања простора претраге прелази се на извршавање quadtree алгоритма, функција која врши рекурзивно партиционисање слике зове се *quadtree*. Ова фунција најпре извршава рекурзивно партиционисање слике до жељене минималне дубине. Када се постигне минимална дубина врши се класификација региона позивом функције classify. У зависности од повратних вредности функције classify врши се упоређивање региона са листом домена исте класификације. У колико је у листи пронађен домен са којим је Еуклитско растојање мање од минимално захтеваног, подаци о домену се памте. У супротном, врши се партиционисање региона све док се не пронађе такав домен, или ако није пронађен такав домен ни на највећој дубини, памти се домен са којим је постигнуто најмање Еуклидско растојање. При сваком партиционисању слике у излазни фајл се уписује јединица како би касније у процесу декомпресије могло реконструисати гранање слике на регионе.

Функција која врши упоређивање региона са доменом зове се *compare*. Повратна вредност ове функције је Еуклидско растојање региона и домена. Такође compare функција израчунава оптимални контраст и осветљење за постизање минималног растојања домена од региона.

Функција којом се врши паковање података компресије у излазни фајл назива се *pack*. Паковање је битско, што значи да се као параметар функцији pack поред податка који се жели уписати у излазни фајл наводи и број бита који је минимално потребан за смештање податка. Када број битова података попуни цео бајт, он се уписује у излазни фајл.

6.4. Фајлови DecR.cpp, DecG.cpp и DecB.cpp

Фајлови потребни за декомпресију фрактално компресоване слике налазе се у фајловима DecR.cpp, DecG.cpp и DecB.cpp у зависности од боје која се декомпресује.

Проћићемо кроз декомпресију само једне боје јер је разлика између фајлова само у називу улазног фајла у коме је смештена компресована слика.

6.4.1. Код за декомпресију

Основне функције које чине код за декомпресију су:

1. unpack
2. read_transformations
3. apply_transformation
4. smooth_image

Функција *unpack* по свему је инверзна функцији pack у коду за компресију. Из бинарног фајла се чита бајт по бајт али функција unpack враћа само онолико битова колико се од ње захтева.

Функција *read_transformations* чита све трансформације смештене у фајлу компресоване слике и пакује их у меморију. Такође она упоредо врши и израчунавање преосталих неопходних параметара за декомпресију. Read_transformations је рекурзивна функција која по својој структури у многоме подсећа на quadtree функцију из кода за компресију слике.

Формирање слике извршава се у функцији apply_transformation применом афиних трансформације предходно смештених у меморији. Функција apply_transformation представља једну итерацију у формирању коначне слике, укупан број итерација је 10.

У циљу углачавања ивица насталих због блоковски заснованог алгоритма фракталне компресије примењује се функција *smooth_image*. Функција се извршава након завршетка свих итерација, чиме се интерполација примењује само једном на већ формираном сликом.

7. ЗАКЉУЧАК

У овом делу биће изложени неки од експерименталних резултата алгоритма коришћеног у овом раду. Затим ћемо се осврнути на постојећу комерцијалну примену фракталне компресије као и области у којима би фрактала компресија требала да доживи потпуну афирмацију.

7.1. Експериментални резултати

Како перформансе алгоритма за фракталну компресију слике у великој мери зависе од начина имплементације и примењене оптимизације, треба имати на уму да резултати изложени у овој одељку не представљају и њене оптималне вредности. Сви подаци који ће бити предочени су добијени директном применом кода овог рада. Перформансе оваквог алгоритма биће упоређене са JPEG која је већ дуже време, како смо рекли, стандард за компресовану слику.

За објективну процену квалитета компримоване слике најчешће се користи PSNR (вршни однос сигнала шум) [4], који је за слику са 256 нивоа сивог дефинисан са:

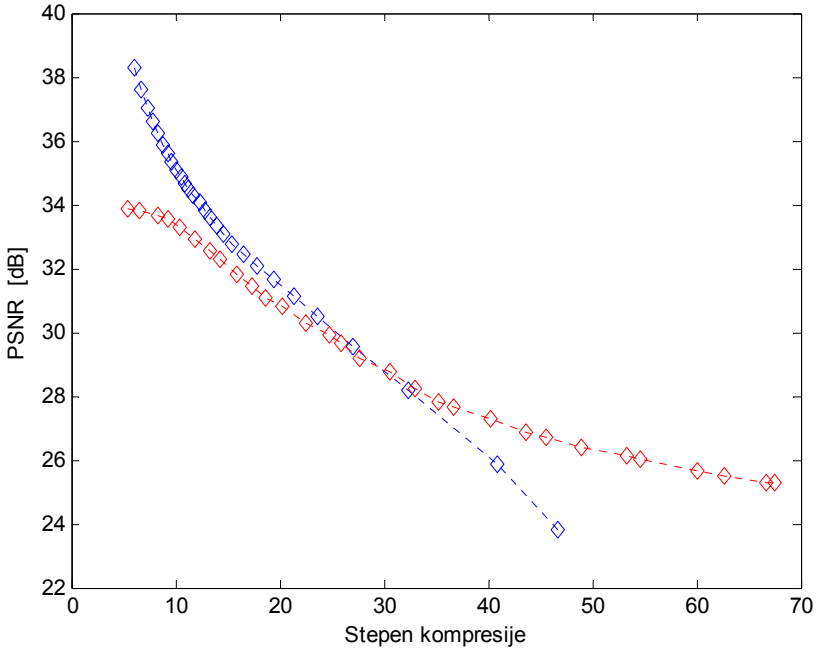
$$PSNR = 10\log\frac{255^2}{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|A[m,n]-B[m,n]\right|^2} \tag{50}$$

Како се врши процена RGB слике, формула (50) добија облик:

$$PSNR = 10\log\frac{255^2}{\frac{1}{3\cdot MN}\sum_{1=1}^{3}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|A[m,n]-B[m,n]\right|^2} \tag{51}$$

У формули (51) са M и N обележен је број пиксела вертикално, односно хоризонтално, а са $A[m,n]$ интензитет пиксел на (m, n) месту некомпресоване слике, док је са $B[m,n]$ означен интензитет пиксела фрактално компресоване слике на месту (m, n).

На Сл. 7.1. су дате вредности PSNR-а при различитим нивоима компресије. Плавом бојом је означена JPEG компресија, а црвеном фрактална. Резултати JPEG компресије формирани су уз помоћ MATLAB-а, док су вредности PSNR за фракталну компресију приказани на слици добијени применом кода овог рада.



Сл. 7.1. *Зависност PSNR од степена компресије*

Са Сл. 7.1. се може видети да за степен компресије мањи од 30 JPEG нуди бољи PSNR, а за већи степен компресије фрактална има боље резултате. Овај закључак се поклапа са нашим очекивањима. Већи степен компресије JPEG-а од 45: 1 није било могуће добити јер није подржан у MATLAB-у. Ово је сасвим разумљиво јер JPEG није ни предвиђен за употребу у случајевима велике комресије. Да би се стекао потпуни осећај разлике фрактална компресије и JPEG-а при високом степену компресије на Сл. 7.2. дата је Ленина слика при компресији од 45:1.

a) JPEG b)FRAC

Сл. 7.2. *Слика Lenna при компресији од 45:1*

Осим тога уочена је и чињеница да се при само малој промени осветљења може значајно побољшати PSNR фрактална компресије.

Постигнути резултати приказани на Сл. 7.1. неће бити исти ако је слика узета са web камере. Наиме, слика web камере садржи велику количину шума, тако да ће резултати фракталне компресије, добијени за мали степен компресије, дати исувише мали PSNR. Декомпресована слика иако малог PSNR-а одаваће често утисак далеко квалитетније слике. Ова привидна контрадикторност, се лако може објаснити чињеницом да PSNR даје само пикселску разлику декомпресоване слике од оригинала, а не и оцену визуелног квалитета слике. У оваквом случају фрактална компресија ће се понашати као нископропусни филтер, који ће ″опеглати″ шум камере.

Ослонимо се још једном на меру субјективне оцене квалитета компресије. Погледајмо Сл. 7.3. на којој се види увећани делови познате слике *Паприке*, степен компресије је за обе слике исти и износи свега 14:1. На сликама је повећан контраст ради лакшег уочавања разлике.

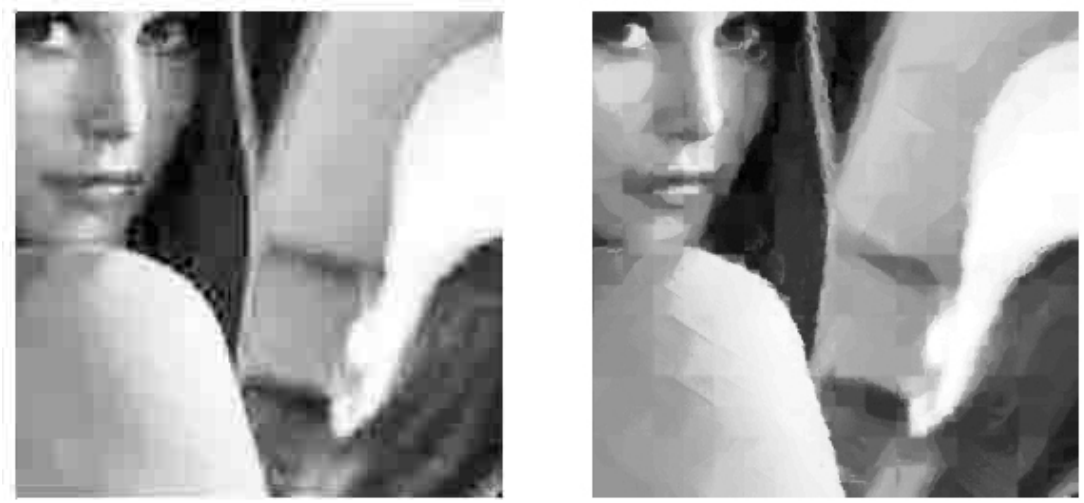
а) JPEG б) FRAC

Сл. 7.3. *Визуелни квалитет слике*

На Сл. 7.3. а) се види да се при већим степенима JPEG компресије долази до изражаја блоковска природа компресије, као последица одбацивања великог броја хармоника. Код фракталне компресије овај проблем је решен постојењам *интерполације* у поступку декомпресије, чиме је далеко поправљен визуелни квалитет компресоване слике.

У уводном делу истакли смо једну важну карактеристику фракталне компресије која се огледа у независности величине компресоване слике и њој одговарајуће декомпресоване. У случају JPEG-а компресован слика величине, рецимо, 256x256, када се декомпресује она је такође величине 256x256 и да би се извршило њено увећање у слику од 512x512, потребно је поседовати неки од програма за обраду слике. Такође, увећана слика има изражену пикселизацију, јер је потребно од једног пиксела формирати четири или више нових. А подсетимо се још једном, фракталном компресијом не чува се информација о пикселима слике, већ се слика описује одговарајућим трансформацијама. Таквим приступом је омогућено да се компресована слике још у самом процесу декомпресије може декомпресовати на било коју величину, а да притом не дође до пикселизације која је се дешава при чистом повећању. Наравно, јасно је да увећана слика неће садржати нове детаље. На Сл. 7.4. приказани су делови

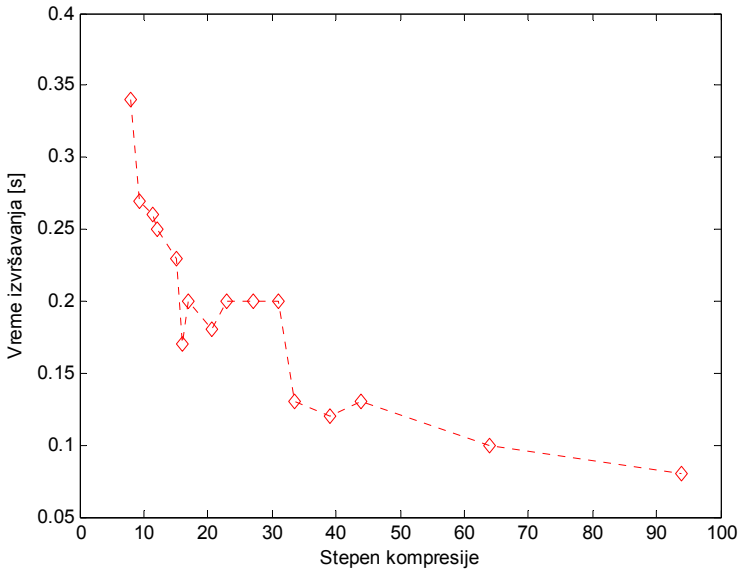
слике Лене која је компресована у величини 256x256, а затим декомпресован. JEPG слика је увећана на 512x512 уз помоћ PhotoShop-а, док је за формирање фракталне слике коришћен допуњен код овог рада који подржава и поменуту опцију. На сликама је повећан контраст ради лакшег уочавања разлике.



a) JPEG b)FRAC

Сл. 7.4. *Пикселизација приликом увећања слике*

Време извршавања фракталне компресије представља једну од њених највећих мана. Како се последњих година интезивно радило на решавању овог проблема, развијени су алгоритми који су се приближили времену компресије JPEG-а.



Сл. 7.5. *Време компресије у зависности од степена компресије*

Време фракталне компресије у зависности од степена компресије изложено је на Сл. 7.5. Код quadtree алгоритма повећањем прага толеранције постиже се већи степен компресије, а смањује се број грањања јер се одговарајући домен брже налази, што све има за последицу краће време извршавања алгоритма. Време JPEG компресије је између 20ms и 10ms на

истом рачунару, па се може закључити да тек при великом степену компресије, преко 30ms, време фракталне компресије бива упоредиво са JPEG-ом.

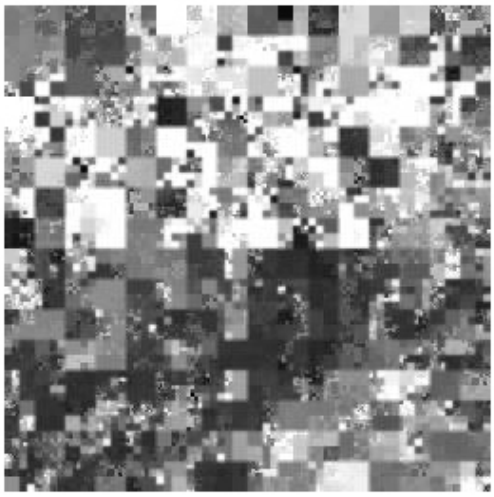
На Сл. 7.5. је приказано време фракталне компресије када је у алгоритму коришћена класификација на основу варијансе. Постојање ове класификације је убрзало компресију преко 10 пута у односу на алгоритам у коме постоји само класификација на основу осветљења. Цена која се морала платити је у степену компресије и квалитету слике. У табели 4. биће изложени упоредни подаци фракталне компресије са и без класификације на основу варијансе неких од познатијих слика које се користе за оцену квалитета компресије.

Слика	Класификација по варијанси	Степен компресије	PSNR	Време
Lenna	не	14.3	32.3	2.46
	да	11.8	30.1	0.24
Камерман	не	11.3	28.5	3.4
	да	10.6	25.1	0.26
Паприке	не	15	33.8	2.6
	да	12	32.2	0.23
Цвеће	не	6.8	28.8	5.5
	да	6.3	25	0.39

Табела 4. Фрактална компресија са и без класификације по варијанси

Quadtree алгоритам иако има велике предности које смо помињали у глави 4.1., крије у себи један велики недостатак. До сада смо посматрали идеалан пренос података од места где се врши компресија до места где се ти подаци декомпресују у слику. Шта ће се десити ако у каналу за пренос података постоји шум и како ће се то одразити на слику? Сетимо се структуре података за трансформацију, дакле преноси се контраст и осветљење, као и положај и орјентација домена, шум над овим подацима ће свако донети изобличење целог региона над којим се врши трансформација. Како се декомпресија обавља итеративним процесом над истом сликом, грешка се кумулативно преноси и на регионе чији је домен претходно оштећени регион. Али ово није најгори случај, наиме, трансформације нису једини подаци који се преносе, рекли смо да при сваком партиционисању слике, у фајл се уписује јединица која означава да се врши даље дељење слике. У колико се шум јави на овом месту и ми погрешно прочитамо бит партиционисања, неће бити само регион на који се партиционисање односи уништен, већ цела слика. На Сл. 7.6. а) приказана је слика Лене када на линији преноса постоји шум који је оштетио неке од података трансформације, а на десној слици је приказано шта се дешава ако је само један бит партиционисања погрешно пренет.

а) шум на битовима трансформације



б) шум на биту грањања

Сл. 7.6. *Слика са присуством шума на линији преноса*

Лек за овакав случај би могао бити и у адекватнијем означавању партиционисања, рецимо уместо једног, три бита јединица, па би се у декодеру одлучивало већински... Али би то свакако довело до неког смањења компресије.

До сада смо се само бавили перформансама алгоритма за компресију слике, погледајмо шта је са декомпресијом. За разлику од JPEG компресије, код које је декомпресија и компресија симетрична у сваком погледу, фрактална компресија и декомпресија су асиметрични процеси. Као што је то у глави 5. објашњено декомпресија је изузетно једноставна и далеко бржа од компресије. Време извршавања декомпресије примењене у овом раду је испод 20ms. Постоје новији алгоритми код којих је и ово време умањено. Чињеница да је алгоритам декомпресије јако једноставан и брз наводи на многе потенцијалне примене фракталне компресије.

7.2. Комерцијална примена

Досадашња истраживања показала су да фрактална компресија поседује квалитет потребан за комерцијалну примену. Карактеристике које је издвајају у односу на друге методе компресије су:

- могућност постпроцесно скалирање величине слике
- велика брзина декомпресије
- висок квалетет слике при великом степену компресије

Увидевши ове закључке 1991. основана је фирма Iterated System, која је прва развила комерцијалне апликације базирану на фракталној компресији. Човек који је стао иза фирме био је управо Michael Barnsley, најзаслужнија особа за приближавање фракталне теорије компресији слике. Задовољна резултатима које је ова фирма постигла Microsoft одлучује да примењује њихов алгоритам за компресију слике у својој чувеној дигиталној енциклопедији Encarta. Шкотска затворска служба га употребљава за стварање дартотеке слика криминалаца, а Queen Mother Library универзитета Aberdeen се послужили фракталном компресијом за електронско складиштење 44000

историјских фотографија. Као екстензија за фрактално компресовану слику коришћена је FIF (**F**ractal **I**mage File **F**ormat), која је и данас једина шире позната екстензија.

Упркос овим изварендним почетним успесима, уследио је пад у заинтересованости за фракталну компресију. Објаснимо, циљ развоја алгоритма фракталне компресије фирме Iterated System је био за његову употребу од стране корисника интернета који су у то време располагали скромним битским протоком (мањим од 56К). Чак је понуђена и могућност да се фрактално компресована слика декомпресује у самом процесу отварања странице. Тако би корисник који би препознао да није отворио жељену слику на нижој резолуцији, могао одмах да се пребаци на другу. Али, наглим развојем интернет комуникација већ средином деведесетих, потреба за великим степеном компресије на штету квалитета опада. JPEG тада успоставља свој примат са одличним квалитетом слике при компресији до 20:1. Он такође поседује још једну, можда у том моменту и најбитнију предност. Наиме, JPEG је индустриски стандард и као такав потпуно бесплатан, док је фрактална компресија заштићена U.S. патентом. Фирма Iterated System никада није објавила своја алгоритам компресије што је такође у великој мери утицало на даљи развој фракталне компресије.

У новије време појавиле су се фирме који су саме развиле своје алгоритме за фракталну компресију. Фирма Altamira Group пласирала је производ Genuine Fractals, који је касније одкупила фирма Lizard Tech. Тренутно актуелна верзија Genuine Fractals је 4.0 и инсталира се као додатак за PhotoShop. Genuine Fractals открива све могућности и предности фракталне обраде слике. Предвиђен је за употребу у области дигиталне обраде фотографије, сателитских снимака и скенираних докумената.

Фрактално генерисана слика постала је стандардни алат за специјалне ефекте у снимању филмова. Познато је да се користило за снимање *Star Wors*-а и *Star Trek*-а. Такође, користи се све више и у индустрији компјутерских игара.

Фирма Iterated System је наставила да ради под именом MediaBin и постизала је одличне резултате у компресији слике комбинујући фракталну компресију са другим методама компресије. Она је такође започела пројекат ClearVideo, који је представљао систем за фракталну компресију видео секвенце. Фирма Real Networks је откупила овај пројекат и преименовала га у RealVideo, тренутно је актуелна верзија RealVideo 10, која има карактеристике:

- Бољи квалитет при више од двоструко мањем протоколу MPEG-2
- Бољи квалитет при двоструко мањем протоколу од MPEG-4
- Компресијом се може сместити 180 мин. видео секвенце на обичан CD
- Компресијом се може сместити 900-1350 мин. видео секвенце на DVD
- Квалитет приближан DVD-у при прококу од 400Kbps
- При протоку од 34Kbps се може преносити видео секвенца 4 пута мање величине слике

Наведен карактеристике задовољавају потребе модемских корисника, а из последње ставке се види да чак и корисници интернета са слабијом конекцијом могу у реалном времену гледати видео секвенце.

Управо у чињеници да фрактална компресија нуди бољи квалитет слике при малим протоцима даје основу за његово коришћење у областима телекомуникација где расположив проток није довољан за употребу других компресија. Овакав случај је са преносом видео секвенце мобилних корисника.

Дакле из свега виђеног, најбољи резултати су се показали у комбиновању фракталне компресије са другим методама, рецимо wavelet компресијом, где су постигнути резултати далеко превазишли конкуренцију. Па је могуће очекивати да као таква фракталана компресија постане саставни део неког будућег MPEG (**M**oving **P**ictures **E**xperts **G**roup) стандарда. Реално је и очекивати експанзију фракталне компресије у условима где је због ограничености пропусног опсега за пренос информација потребан велики степен компресије, као што је то случај са мобилном телефонијом, сателитским фотографијама,....

8. ЛИТЕРАТУРА

[1] Michael Barnsley: *Fractal Everywhere*, Academic Press, Inc., 1988.

[2] Милан Меркле: *Математичка анализа, теорија*, Друго издање, Академска мисао, 2001.

[3] Д. М. Цветковић, И. Б. Лацковић, М. Ј. Меркле, З. С. Радосављевић, С. К. Симић, П. М. Васић: *Математика I, алгебра*, Седмо издање, Академска мисао, 2000.

[4] Миодраг Поповић: *Дигитална обрада слике*, необављена књига, http://tnt.etf.bg.ac.yu

[5] Mark Nelson: *The Data Compression Book*, Secon edition, John Wiley & Sons Canada, Ltd, 1995.

[6] Ласло Краус: *Програмски језик C са решеним задацима*, Академска мисао, 2006.

[7] *Getting started with MATLAB*, Version 7., The MathWork

[8] Kenneth Felconer: *Fractal geometry, Matematical foundations and applications*, Second edition, ohn Wiley & Sons, Ltd., 2003.

[9] David J. Wright, *Dynamical Systems and Fractal Lecture Notes*, електронска књига

[10] John Kominek: *Understanding Fractal Image Compression*,

[11] D. Soupe, R. Hamzaoui, H. Hartenstein: *Fractal image compression an introduction overview*

[12] Yuval Fisher: *Fractal Image Compression*

[13] John Kominek: *Advances in Fractal Compression for Multimedia Applications*

[14] Yuval Fisher: *A Comparison of Fractal Methods with DCT and Wavelets*

[15] Nguyen Ky Giang, Dietmar Saupe: *Adaptive post-processing for fractal image compression*

[16] John Kominek: *Restoration of PIFS Encoded Images*

[17] Душан Грујић: *Фрактална компресија слике*, семинарски рад

.[18]http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=247 сајт са кога се може преузети Vision for MATLAB (VFM)

[19] http://links.uwaterloo.ca/ , Линк Универзитета Waterloo

[20]. http://spanky.triumf.ca/ , The Spanky Fractal Database, историјат фрактала и остале корисне информације

[21] http://web.math.hr/~mathe/frakdim/index.html#Fraktalna , Хрватски математички електронски часопис

[22] http://www.uni-giessen.de/faq/archiv/sci.fractals-faq/ , Сајт Универзитета Giessen, FAQ о фракталима

[23] http://www.faqs.org/faqs/compression-faq/part2/section-8.html , Introduction to fractal compression, Internet FAQ archives

[24] http://www.compression-links.info/Fractal , Сајт са корисним линковима

[25] http://inls.ucsd.edu/~fisher/Fractals/ , Сајт Yuval Fisher-a

[26] http://www.lizardtech.com/press/news.php?item=02-25-2005 , LizardTech сајт, Geniune Fractal 10

[27] http://www.atp.nist.gov/eao/sp950-3/mediabin.htm , извештај ATP-a о резултатима улагања у фирму Iterated Systems, Inc.

[28] http://www.real.com/ , Сајт фирме Real Networks