

ELEKTROTEHNIČKI FAKULTET

OE4DOS

Drugi domaći iz digitalne obrade slike

Autor:

Vuk VUKOMANOVIC 18-2014



Contents

1	Uvod	2
2	Poboljšavanje kvaliteta digitalizovane štampane slike	3
3	Decimacije slike	10
4	Uklanjanje degradacije nastale usled pokreta	14
5	Non local means algoritam	20
5.1	Implementacija	22
5.2	Rezultati	23

1 Uvod

Izveštaj je podeljen u logičke celine, koje se odnose na taskove koji su zadati drugim domaćim zadatkom. U izveštaju se nalaze slike i rezultati koji su generisani mimo poslatih .m fajlova radi manje robosnosti. Shodno tome ceo dokument je podeljen u sledeće četiri celine:

1. Poboljšavanje kvaliteta slike **girl_print.jpg**
2. Realizacija funkcije dos_downscale i poređenje sa ugrađenim funkcijama
3. Uklanjanje degradacije nastale usled pokreta kamere
4. Realizacija algoritma ne-lokalnog usrednjavanja

2 Poboljšavanje kvaliteta digitalizovane štampane slike

Na slici ispod, Figure1, je prikazana ulazna slika čiji kvalitet je potrebno poboljšati.



Figure 1: Originalna slika

Vizuelnom inspekcijom slike se primećuje da je prisutan Moirov efekt koji je nastao pri stampanju ove slike. Najverovatnije je da je ova slika iz novina

pošto se ovom tehnikom dosta smanjuje količina potrebne boje za stampanje istih. Kako bi odstranili ovu smetnju prvo je potrebno posmatrati spektar ove slike, koji je dat u nastavku.

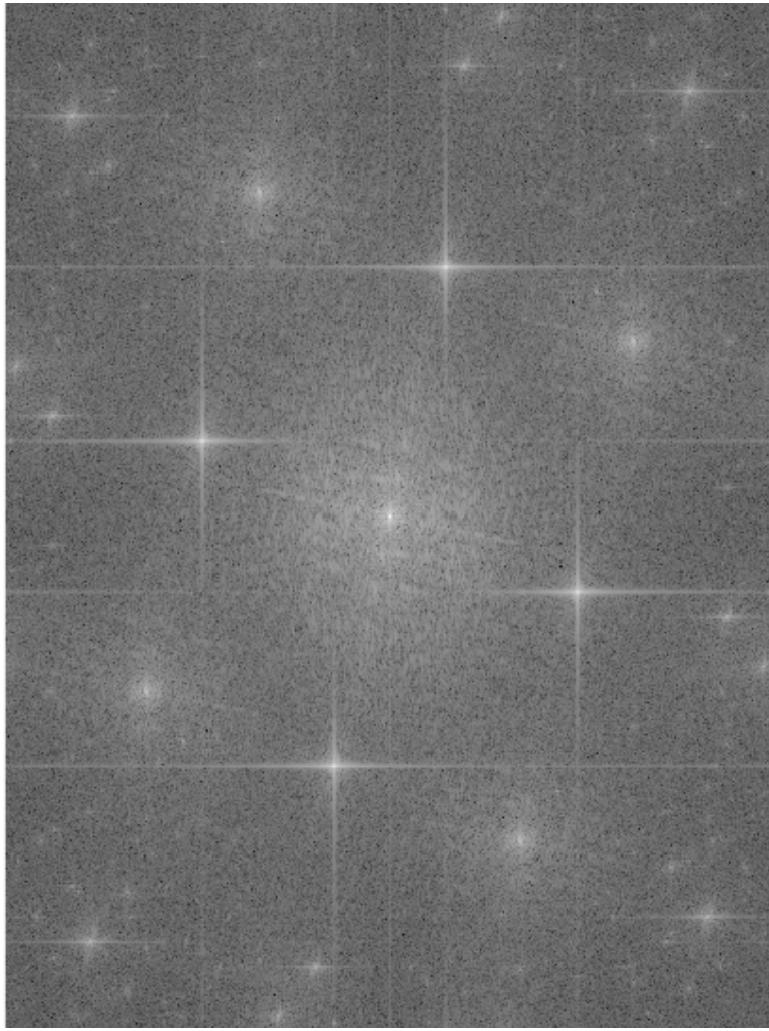


Figure 2: Spektar ulazne slike

Na spektru, Figure 2, se jasno vide komponente koje prave Moirov efekat. Ove komponente se mogu isfiltrirati noć filtrom. U kodu se mogu videti koordinate centara komponenti koje su izbačene. Na slici ispod je napravljeni noć filter.

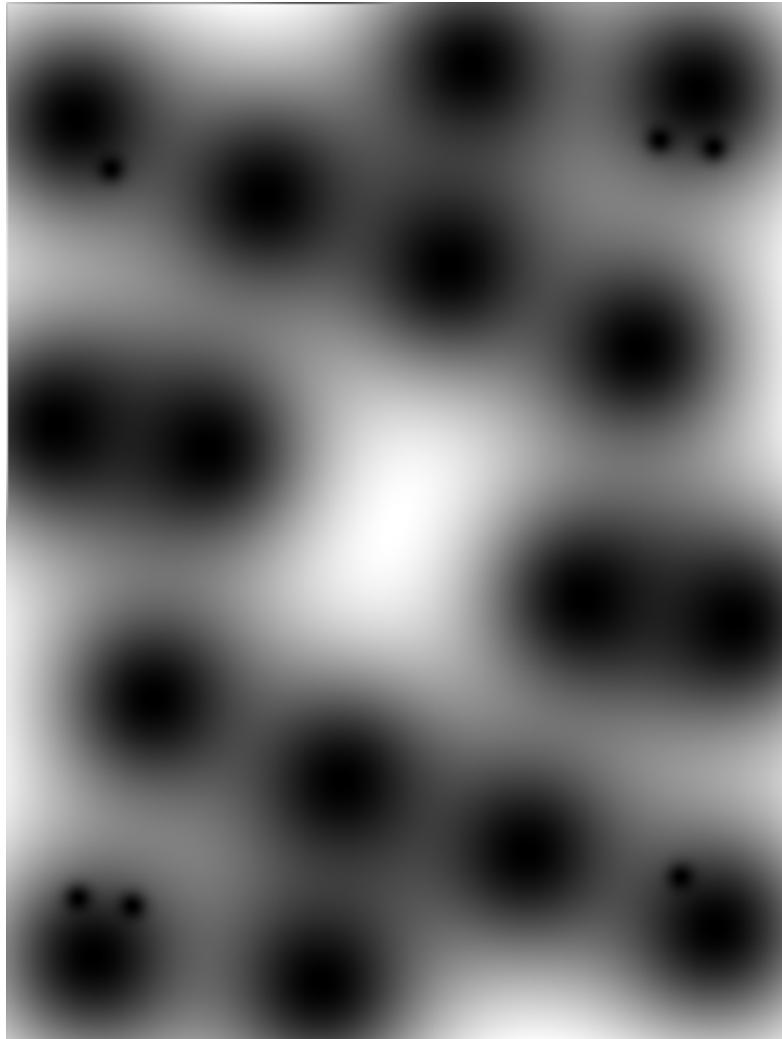


Figure 3: Spektar noč filtra

Na spektru fitlra se vide veći i manji krugovi, manjim krugovima je pokušano da se sačuva što više originalnog spektra slike. Nakon filtriranja ima se spektar:

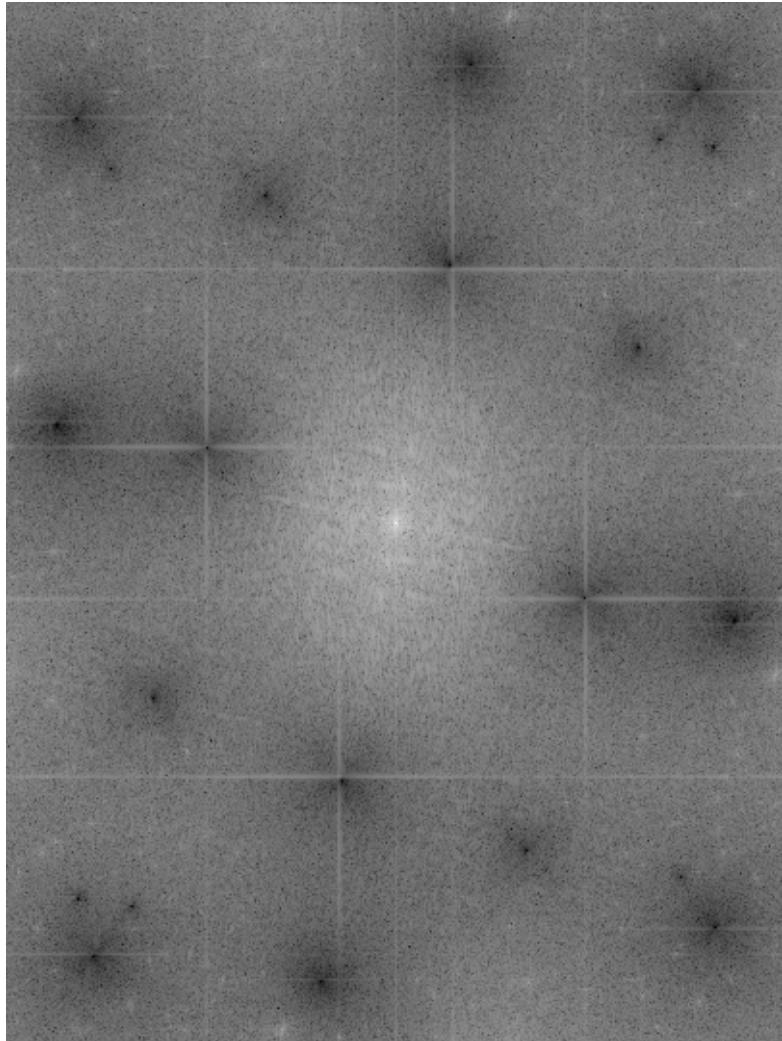


Figure 4: Spektar noč filtra

Na slici iznad je spektar nakon filtriranja, jasno se vidi da su „krugovi,, isfiltrirani ali da od nekih jačih komponenti su ostale horizontalne i vertikalne komponente. Zbog njih dodajemo vertikalne i horizontalne noč filtre. Spektar nakon primene tih filtra je dat u nastavku.

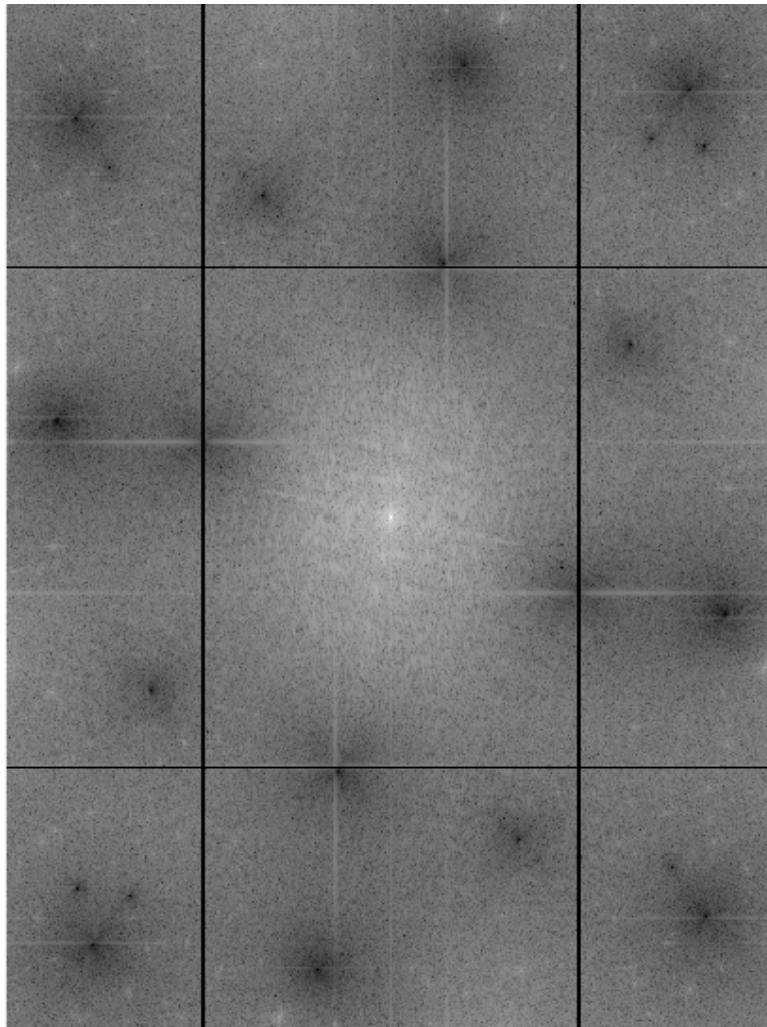


Figure 5: Spektar nakon primene vertikalnog i horizontalnog noč filtra

Nakon ovih promena u frekvencijskom domenu, spektri su vraćeni u prostorni domen inverznom Furijeovom transformacijom. Na prvoj slici, Figure 6, je izlazna slika nakon primene samo kruznih noč filtera, a na drugoj, Figure 7, je nakon primene svih filtera.

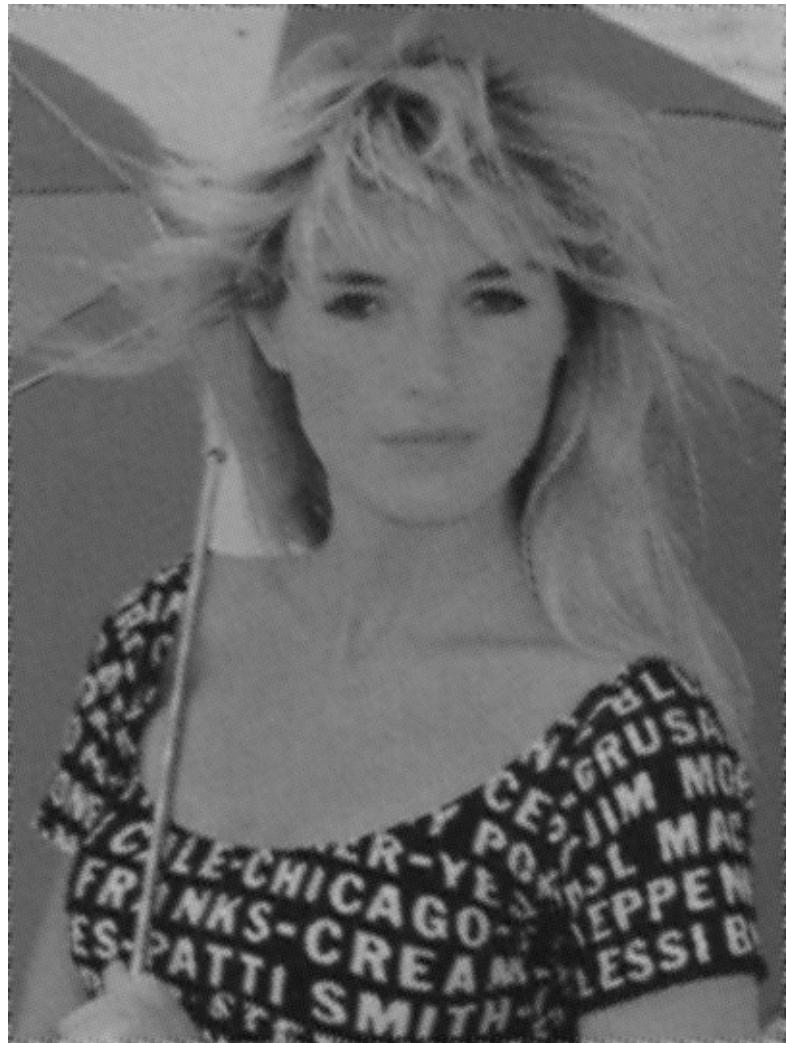


Figure 6: Izlzna slika

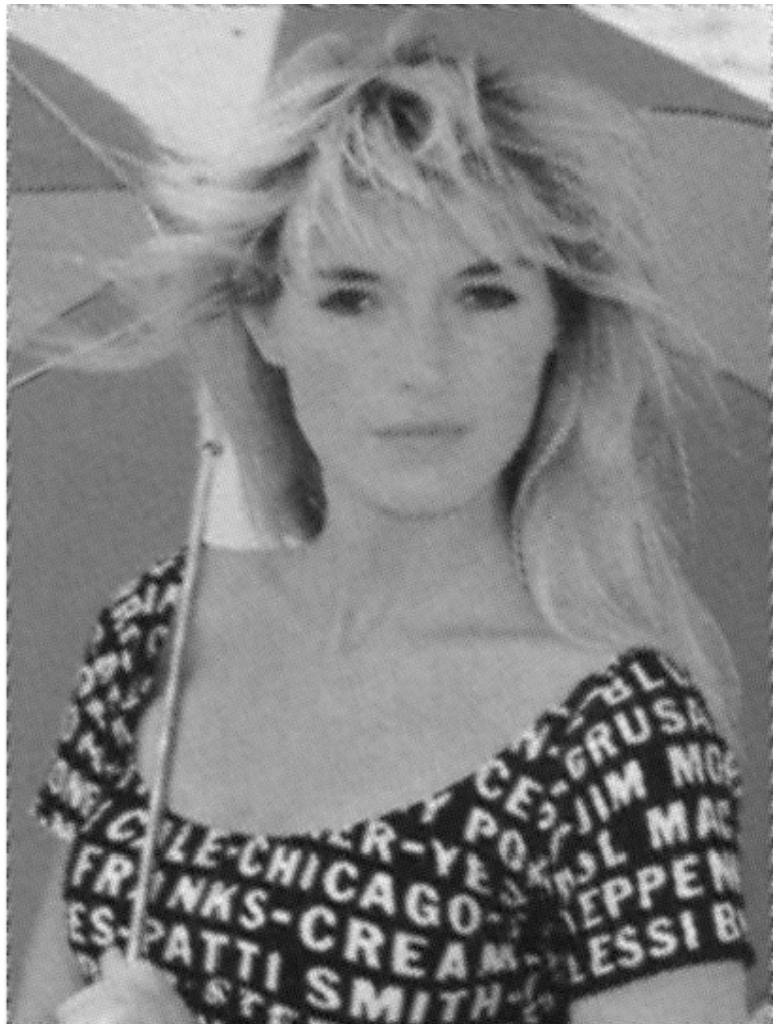


Figure 7: Slika nakon primene vertikalnog i horizontalnog noč filtra

Sa ove dve slike se primećuje da je Moirov efekat u velikoj meri isfiltriran. Kvalitet slike može dodatno da se poboljša isticanjem ivica i poboljšanjem kontrasta, ali pošto u postavci domaćeg nije napisano da li je to dozvoljeno, rezultati nisu uključeni u izveštaj.

3 Decimacije slike

Drugi deo domaćeg zadatka se odnosi na decimaciju slike. Kao što je i traženo, napisana je funkcija dos_downscale koja vrši decimaciju ulazne slike metodom najbližeg suseda. Metod najbližeg suseda je jednostavna metoda za menjanje dimenzija slike, prema njoj vrednost izlaznog piksela odgovara vrednosti najbližeg piksela u originalnoj slici. To se matematički može zapisati sa

$$g(x', y') = f(\text{round}(x), \text{round}(y))$$

Napisana funkcija pre same decimacije ulaznu sliku filtrira low pass filtrom u frekvencijskom domenu, kako bi se spremio aliasing efekat. Ukoliko kažemo da je faktor decimacije s i dimenzije ulazne slike $M \times N$, tada je potrebno filtrirati sliku filtrom sa cut-off frekvencijom ne manjom od $\frac{\max(M, N)}{2s}$. Razlog za ovu frekvenciju je sledeći, Nikvistova frekvencija ulazne slike je $\frac{\max(M, N)}{2}$, a pošto smo efektivno promenili frekvenciju odabiranja smanjenjem dimenzije slike za faktor s , takođe će se i Nikvistova frekvencija promeniti.

Svi rezultati su predstavljeni na slici roof.png.

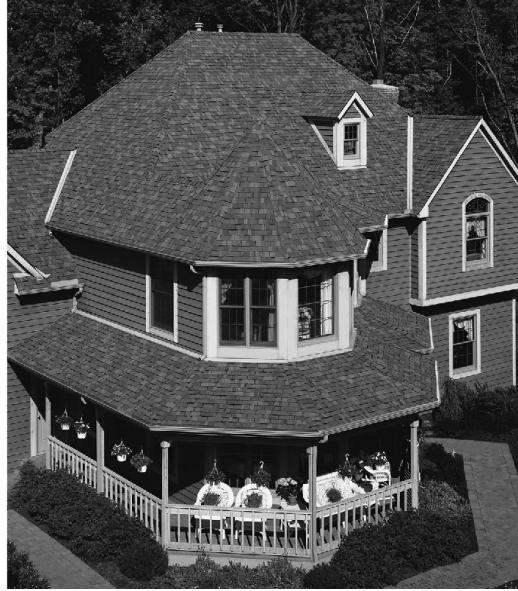


Figure 8: Ulazna slika, roof.jpg

Fun fact : Latex kada skalira slike za prikazivanje ne koristi low pass filter, što se može videti na slici iznad (krov).

U nastavku su dati rezultati funkcije dos_downscale i imresize sa decimacionim faktorom 2,4 i 7, sa i bez anti-aliasing filtra.



Figure 9: dos_downscale

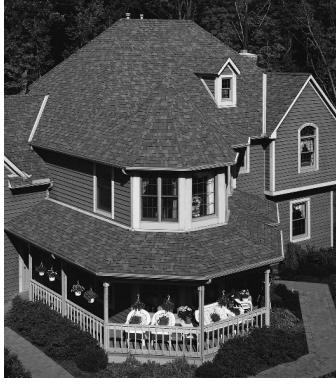


Figure 10: imresize, nearest neighbour



Figure 11: imresize, bicubic

Figure 12: Faktor decimacije 2, anti-alliasing off



Figure 13: dos_downscale



Figure 14: imresize, nearest neighbour



Figure 15: imresize, bicubic

Figure 16: Faktor decimacije 2, anti-alliasing on



Figure
dos_downscale

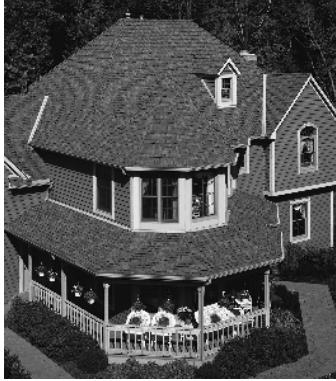


Figure 17: dos_downscale
Figure 18: imresize,
nearest neighbour



Figure 19: imresize,
bicubic

Figure 20: Faktor decimacije 4, anti-alliasing off

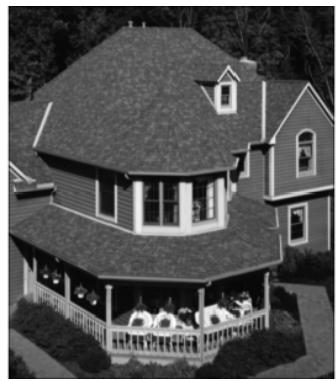


Figure
dos_downscale



Figure 21: dos_downscale
Figure 22: imresize,
nearest neighbour

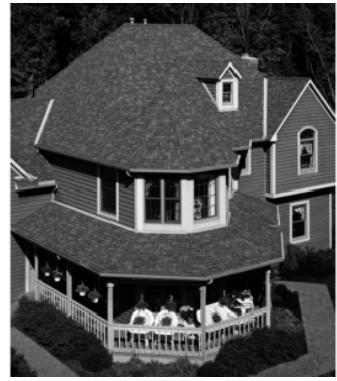


Figure 23: imresize,
bicubic

Figure 24: Faktor decimacije 4, anti-alliasing on

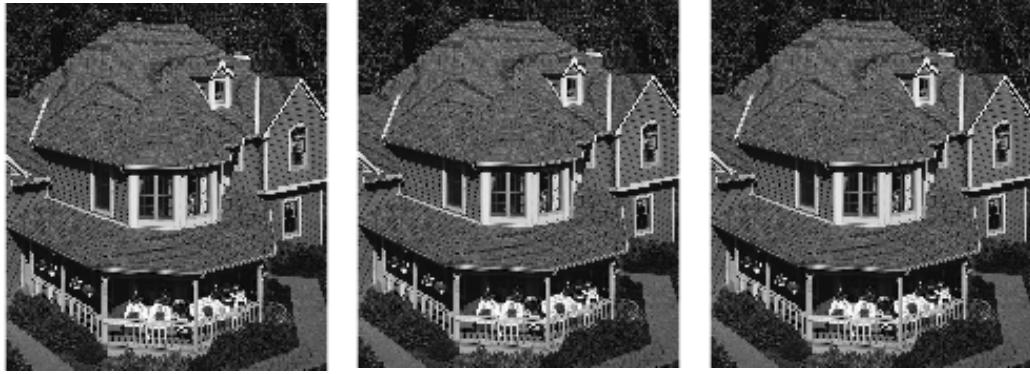


Figure
dos_downscale

25: Figure 26: imresize,
nearest neighbour

Figure 27: imresize,
bicubic

Figure 28: Faktor decimacije 7, anti-alliasing off



Figure
dos_downscale

29: Figure 30: imresize,
nearest neighbour

Figure 31: imresize,
bicubic

Figure 32: Faktor decimacije 7, anti-alliasing on

Kao što je i očekivano bikubična transformacija daje bolje rezultate. Tj. sa prve i druge slike se vidi da pri faktoru decimacije 2, sve 3 metode daju dobre rezultate, tj. efekti aliasinga nisu vidljivi(Figure 18,19). Za faktor decimacije 4 na krovu kuce se vidi aliasing efekat za metodu najbližeg suseda (custom i matlabova funkcija, Figure 20). Na narednoj slici se vidi da je filtriranje uspesno(Figure 24). I na kraju za decimacioni faktor 7 bez anti-aliasing filtra kod izlaza sve 3 metode je prisutan aliasing efekat,na krovu kuce(Figure 28), takođe na narednoj slici (Figure 32), se vidi da filter sprečava preklapanje u spektru. Treba napomenuti da iako low pass filter sprečava preklapanje spektralnih komponenti, takođe i zamenuje sliku.

Note Za ispravno vidjenje rezultata potrebno je pokrenuti skriptu, pošto se iz ovog pdf-a jako malo vidi.

4 Uklanjanje degradacije nastale usled pokreta

Treći deo domaćeg se odnosi na uklanjanje degradacije nastale usled pokreta kamere **ako** je sam pokret poznat. Ulazna slika kao i sam pokret su dati u nastavku repsektivno.



Figure 33: Ulazna slika, etf.jpg

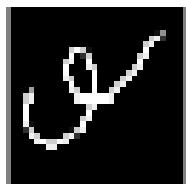


Figure 34: Pokret

Pošto je poznata degradacija koja je napravila problem, rešavanje se svodi na inverzno filtriranje. Naravno kao i uvek kod inverznog filtriranja šum predstavlja problem. Trivijalno rešenje je da se low pass filtrom ograniči opseg i na taj način spreči „pojačanje“, šuma. Drugi način je da se koristi Vinerov filter. Vinerov filter kao kriterijum koristi srednju kvadratnu grešku, tj. teži da je minimizuje. Za ovako nešto je potrebno znati spektar snage šuma i spektar nedegradirane slike što je u ovom slučaju nedostupno, proizvod ove 2 veličine se može uzeti kao jačina filtriranja. Empirijski je izračunata ova vrednost. U nastavku su dati svi koraci.

Kao što je već rečeno prvo je izvršeno inverzno filtriranje uz Vinerov filter.

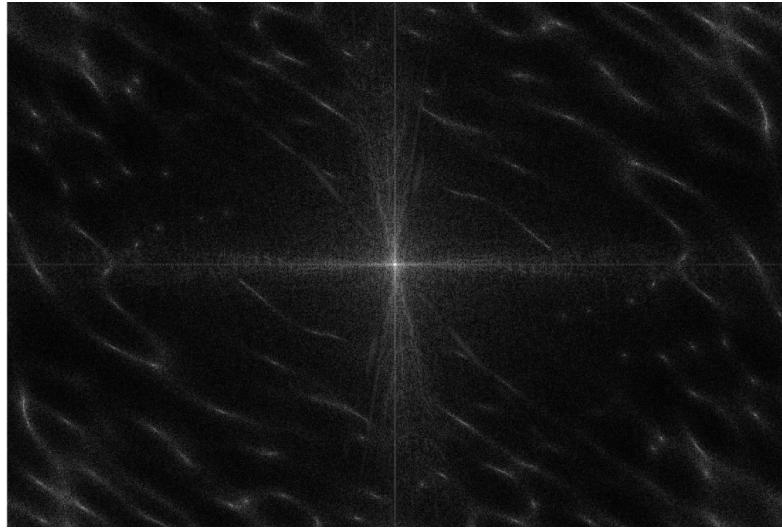


Figure 35: Spektar nakon inverznog filtriranja

Na slici iznad, Figure 35, se nalazi spektar nakon inverznog filtriranja uz Vinerov filter. Na njemu se primećuje da i dalje postoji dosta šuma. Što potvrđuje i slika koja odgovara ovom spektru, Figure 36.



Figure 36: Slika nakon inverznog filtriranja

Sa slike se vidi da je degradacija usled pokreta uspešno isfiltrirana ali da slika sada ima dosta šuma na njoj. Posmatrajući spektar, Figure 35, primen-

jene su različite metode.

Prva metoda koje je primenjena jeste prost low pass filter. Spektar ovog filtra kao i rezultat filtriranja su dati u nastavku.

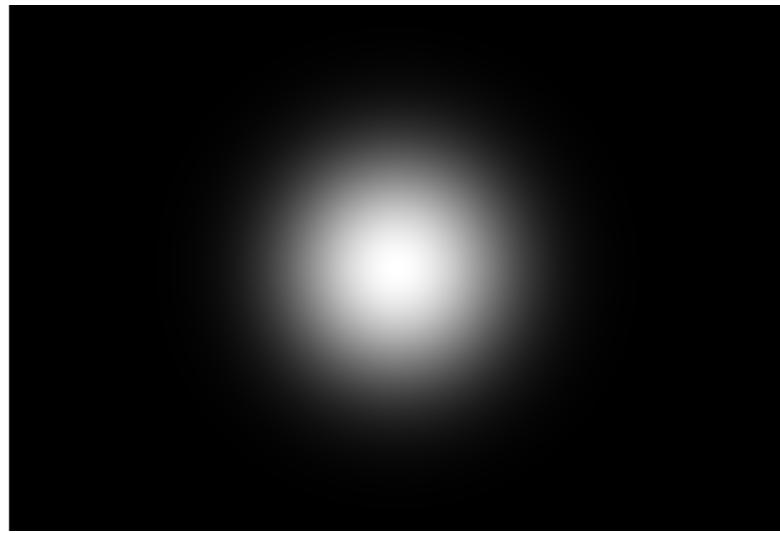


Figure 37: Spektar korišćenog low pass filtra

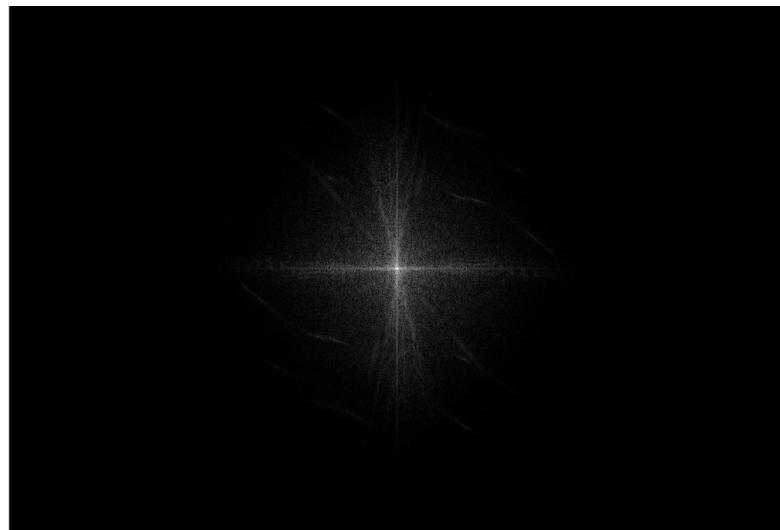


Figure 38: Spektar nakon low pass filtriranja



Figure 39: Slika nakon low pass filtriranja

Primećuje se da je ova metoda isfiltrirala deo šuma ali je isto tako i zamutila sliku, što je bilo i očekivano.

Druga metoda koja je korišćena je adaptivno lokalno usrednjavanje. Sa uniformnog dela slike je određena varijansa šuma koja je koršnjena pri realizaciji algoritma.

Rezultat je dat u nastavku.



Figure 40: Slika nakon low pass filtriranja

Treća metoda je inspirisana time da se low pass metodom izgubi dosta komponenti koje se nalaze na pravama koje se nalaze u sredini, vertikalno i horizontalno. Napravljen je „krst,, pass notch filter ali nije dao zadovoljavajuće rezultate pa nije uključen u izveštaj.

5 Non local means algoritam

Non local means algoritam predstavlja jednu od metoda za filtriranje slike. Kao što samo ime kaže, za razliku od uobičajnih algoritama usrednjavanja, non local means ne gleda geometrijski bliske piksele već slične piksele. Dva piksela su slična ako njihove okoline liče. Algoritam će biti objašnjen uz pomoć naredne slike. Kao što se na ovoj slici može primetiti prirodne slike imaju dosta sličnosti na sebi, tako na primer su okoline A i B veoma slične za razliku od B i C, takođe D i E su veoma slične ali se razlikuju od ostalih spomenutih okolina. Ideja non local means algoritma je da se ne ograničavamo na usrednjavanje neke okoline , nego na usednjavanje sličnih piksela.



Figure 41: Primer

Svaki piksel na slici je opisan svojom okolinom, tj. on je centar kvadrata $K \times K$. Ako prepostavimo da na slici postoji šum η , tada svaki piksel i slike

P možemo da predstavimo kao

$$P(i) = P(i)_{signal} + \eta_1$$

Ako zarad objašnjena pretpostavimo da na slici postoje identične okoline ovoj tada imamo

$$P_2(i) = P(i)_{signal} + \eta_2$$

$$P_3(i) = P(i)_{signal} + \eta_3$$

...

$$P_n(i) = P(i)_{signal} + \eta_n$$

. Lako se vidi da , pošto je šum random promenljiva, piksel $P(i)$ je svuda isti, samim tim usrednjavem samo ovih „istih“, piksela vrednost signala se ne menja već se samo usrednjava vrednost šuma, tj. ona se potiskuje. Pošto na slici veoma retko postoje identične okoline uvodi se težinska funkcija koji nosi informaciju o tome koliko su okoline slične, i vrednosti piksela u izraz za usrednjavanje ulaze pomnožene ovom funkcijom. Za jedan piksel na slici određuje se težinska funkcija za sve ostale na slici. Ukoliko je okolina identična ova funkcija ima vrednost 1 , a ukoliko nema nikakvih sličnosti ona teži nuli. Izbor težinske funkcije dosta utiče na sam rad sistema, u ovom domaćem kao težinska funkcija korišćena je Euklidska distanca. Stvar koju je bitno napomenuti jeste da se funkcija mora normalizovati kako bi se ispunio uslov

$$w(p, q) \Rightarrow 0 \wedge w(p, q) \leq 1,$$

gde su p i q pikseli slike.

Nakon uvođenja ove funkcije izraz za piksele izlazne slike se može zapisati

$$NL(i) = \sum w(i, j)v(j)$$

Glavni problem ovog algoritma je kompleksnost, naime za nalazenje sličnih okolina piksela potrebno je konstantno prolaziti kroz celu sliku što je veoma vremenski skupo. Zato se uvođe okolina $S \times S$, koja predstavlja okolinu u kojoj se traže slični pikseli na ovaj načine je smanjena kompleksnost programa na

$$O(n) = MNS^2K^2$$

, gde su M i N dimenzije slike. Ova kompleksnost je i dalje jako velika ali za razliku od pretrazivanje cele slike je za veće dimenzije prihvatljiva.

5.1 Implementacija

Pošto se kompleksnost algoritma ne može smanjiti za zadate parametre, optimizacija se svodi na optimizaciju u određenom jeziku. Pošto matlab tako dobro radi sa vektorima a loše sa for petljama kod je kucan kako bi se for petlje što više smanjile. Sama implementacija je objašnjena u nastavku.

Prvi korak je proširiti ulaznu sliku, kako bi i ivični pikseli bili uključeni u obradu. Slika je proširena za $pad = \frac{S}{2} + \frac{K}{2}$.

Dve neizbežne for petlje su petlje koje idu kroz celu sliku(originalnu, ne proširenu), pošto su ugnježdene njihova kompleksnost je MN . Ostatak funkcije je implementiran na 2 načina. Prvi način je da se sa dva fora za svaki piksel slike obilazi okolina $S \times S$ i vektorski računaju težinske funkcije za trenutni piksel, drugi način je da se sa dva fora služe za brojanje okoline $K \times K$ a da se vektorski računaju težinske funkcije po okolini $S \times S$. Naravno for petlje su botle neck, tako da pristup sa kraćim for petljama će biti brži a to je drugi način.

Pretpostavićemo da smo unutar prve petlje, tj. da su na x i y koordinate trenutnog piksela. Spomenute for petlje se kreću od 0 do $K-1$, tako da u svakom prolazu za jednu poziciju u lokalnom susedstvu $K \times K$, računamo Euklidsko rastojanje sa svim okolinama $K \times K$ u okolini sa pretraživanje $S \times S$. Praktično to znači ako su nam petlje u stanju $(0,0)$ i $K = 3$, tada računamo Euklidsko rastojanje tačke $(x - 1, y - 1)$ sa svim tačkama lokalnih susedstava na toj poziciji, tj. sa $(x_{sn} - 1, y_{sn} - 1)$ gde n predstavlja redni broj piksela na slici za koji računamo težinsku funkciju.

Nakon izlaska iz spomenutih petlji imamo izračunata sva euklidska rastojanja trenutnog posmatranog piksela sa svim kvadratima $K \times K$ u okolini $S \times S$. Sve što je ostalo je da se izručuju težinske funkcije koristeći izraz

$$w(p, q) = e^{-\frac{E(p, q) - \sigma^2}{h^2}}$$

Parametar h kontroliše pad eksponencijalne funkcije i samim tim promenu težinskih funkcija koje su funkcija Euklidske distance. Tj. parametar h predstavlja jačinu filtriranja.

Poslednji korak jeste da se izračuna vrednost piskela izlazne slike koristeći ranije definisanu jednačinu.

$$NL(i) = \sum w(i, j)v(j)$$

Note: Pored potrebnih fajlova priložena je i skripta koja radi sa petljama koje idu po oblastima SxS

5.2 Rezultati

Empirijski određena varijansa šuma je 0.0069 . Za jačinu filtriranja je uzeta vrednost $h = 0.6$ zbog dobijanja najboljeg PSNR-a od svih ispitanih vrednosti. Svi prikazani rezultati su dobijeni filtriranjem slike **lena_noise.png**. U nastavku su date originala slika, zašumljena slika i slika nakon non local means funkcije, respektivno.



Figure 42: Originalna slika



Figure 43: Zašumljena slika



Figure 44: Slika nakon filtriranja, K=9 S=15

Funkcija je testirana za 9 različitih kombinacija parametara K i S. U nastavku je dat grafik zavisnosti PSNR-a filtriranih slika od ulaznih parametara filtriranja.

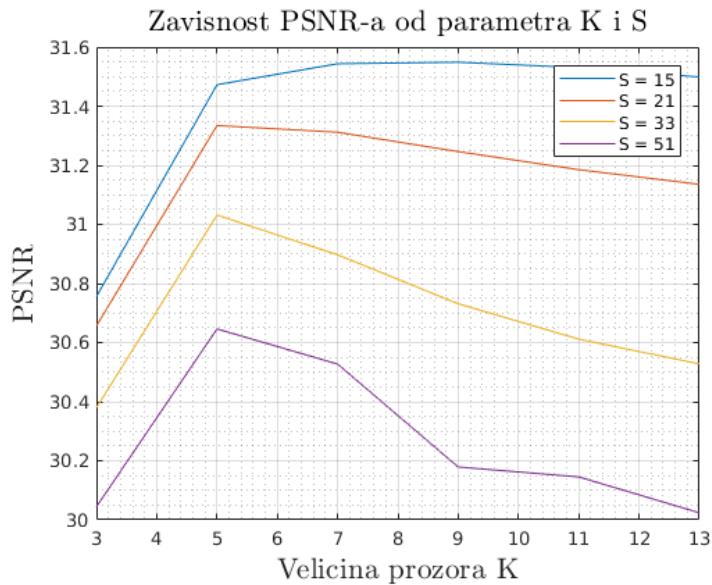


Figure 45: Grafik zavisnosti

Vrednost PSNR-a zasumljene slike, figure 43, iznosi 21.58 dB. Dok vrednost PSNR-a filtrirane slike iznosi 31.55 dB, sto jasno predstavljanje unapredjenje ulazne slike.