

Mladi za napredek Maribora 2015
32. srečanje

Matematična obdelava slik

raziskovalno področje: *matematika*
raziskovalna naloga

Avtor: DOMEN VAUPOTIČ
Mentor: JELKA VOGRINEC
Šola: II. GIMNAZIJA MARIBOR

Maribor, februar 2015

Mladi za napredek Maribora 2015
32. srečanje

Matematična obdelava slik
raziskovalno področje: *matematika*
raziskovalna naloga

Maribor, februar 2015

Povzetek

V raziskovalni nalogi je opisano zajemanje digitalnih fotografij in njihovo matematično predstavljanje z matrikami. Predstavljene so osnovne intenzitetne transformacije, kot so negativ, sprememba svetlosti, razteg kontrasta in upragovanje. Z razlago matematične operacije konvolucije so podana nekatera konvolucijska jedra, s katerimi lahko sliko zameglimo, izostrimo ali poiščemo mejne robove prikazanih objektov. Večji poudarek je na matematični izpeljavi Fourierjeve transformacije, ki omogoča frekvenčno filtriranje slike, in njenem prikazu v obliki digitalne slike; opisana je tudi izpeljava hitre Fourierjeve transformacije (FFT), ki optimira število računskih operacij za izračun transformacije. Uspešno opravljen cilj naloge je bila izdelava računalniškega programa, s katerim lahko v nalogi opisane metode in algoritme preizkusimo tudi na poljubnih digitalnih fotografijah. Nekaj primerov obdelanih slik se nahaja na koncu naloge, program pa je prosto dostopen tudi na spletu.

Profesorici. In minljivosti.

Hvala.

Kazalo

1	UVOD	9
1.1	Cilji naloge	9
1.2	Metodologija	9
2	TEORETIČNI DEL	10
2.1	Osnove digitalne obdelave slik	10
2.1.1	Uporabnost digitalne slikovne obdelave	11
2.2	Zajemanje in predstavljanje slike	13
2.3	Upravljanje z barvami	15
2.3.1	Sivinjenje slike	15
2.4	Operacije med dvema slikama	16
2.5	Predstavitev s histogramom	16
2.6	Intenzitetne transformacije	17
2.7	Obdelava v prostorski domeni	19
2.7.1	Konvolucija	19
2.7.2	Maskiranje	24
2.8	Predstavitev v frekvenčni domeni	24
2.8.1	Fourierjeva vrsta	24
2.8.2	Fourierjeva transformacija	31
2.8.3	Interpretacija Fourierjeve transformacije in filtriranje	35
2.8.4	Nizkoprepustni filter	36
2.8.5	Visokoprepustni filter	36
2.8.6	Pasovnozaporni in pasovnoprepustni filtri	37
2.8.7	Poljubno filtriranje	37
3	RAČUNALNIŠKA IMPLEMENTACIJA	38
3.1	Računanje Fourierjeve transformacije	39
3.1.1	Hitra Fourierjeva transformacija (FFT)	40
3.2	Osnovne procedure	46
3.2.1	Prikazovanje Fourierjeve transformacije	47
3.2.2	Diagrami potekov nekaterih algoritmov	52
3.3	Primeri s programom obdelanih slik	54
3.3.1	Sivinjenje slike	54
3.3.2	Intenzitetne transformacije	55
3.3.3	Obdelava v prostorski domeni	57
3.3.4	Obdelava v frekvenčni domeni	61
4	ZAKLJUČEK	66

5	DRUŽBENA ODGOVORNOST	66
6	VIRI IN LITERATURA	67

Slike

1	Vremenska slika oblačnosti nad Evropo, posneta v infrardečem spektru	12
2	Računalniško generirana slika kvadrov v perspektivni projekciji . . .	12
3	Galaksija Sombrero (M104), posneta z vesoljskim teleskopom Hubble	12
4	Prečna rezina prsnega koša, posneta z računalniško tomografijo (CT)	12
5	Rentgenski posnetek prsnega koša	12
6	Endotelijske celice pljučne arterije goveda, posnete s fluorescenčno mikroskopijo	12
7	Zajemanje digitalne slike z digitalnim fotoaparatom	14
8	Slika žabe	17
9	Histogram slike žabe	17
10	Histogram svetle slike	17
11	Histogram temne slike	17
12	Histogram slike z nizkim kontrastom	17
13	Histogram slike z visokim kontrastom	17
14	Identična transformacija	18
15	Transformacija negativa	18
16	Transformacija svetljenja	18
17	Transformacija temnenja	18
18	Transformacija raztega kontrasta, $\mu = 50$ in $\chi = 2$	19
19	Transformacija raztega kontrasta, $\mu = 200$ in $\chi = 0,8$	19
20	Transformacija raztega kontrasta s sigmoidno funkcijo	19
21	Transformacija upragovanja	19
22	Konvolucija digitalne slike	21
23	Kvadratasti	27
24	Graf funkcij $f(x) \cos(x)$ in $f(x) \cos(2x)$	27
25	Graf funkcij $f(x) \sin(x)$ in $f(x) \sin(2x)$	28
26	Graf Fourierjeve vrste pravokotnega signala	29
27	Graf pravokotnega impulza	32
28	Graf periodičnega pravokotnega impulza, $p = 4$	32
29	Izračunani Fourierjevi koeficienti c_n periodičnega pravokotnega impulza z osnovnimi periodami 2, 4 in 6.	33
30	Skalirani Fourierjevi koeficienti pri $p = 2$	34
31	Skalirani Fourierjevi koeficienti pri $p = 4$	34

32	Skalirani Fourierjevi koeficienti pri $p = 6$	34
33	Fourierjeva transformacija $\hat{\Pi}$ pravokotnega impulza	34
34	Zaslonski posnetek programa	38
35	Vseh 5 kompleksnih rešitev enačbe $z^5 = 1$	41
36	DFT osmih elementov	43
37	FFT osmih elementov	44
38	DFT dveh elementov	44
39	Graf faktorja izboljšave u hitre Fourierjeve transformacije v odvisnosti od števila elementov M	45
40	Centriranje Fourierjeve transformacije	49
41	Periodičnost Fourierjeve transformacije	49
42	Magnituda in faza Fourierjeve transformacije	50
43	Idealni nizkoprepustni filter velikosti 512×512 , $r = 80$	50
44	Idealni visokoprepustni filter velikosti 512×512 , $r = 80$	50
45	Idealni pasovnoprepustni filter velikosti 512×512 , $r = 80$, $r_2 = 130$	50
46	Idealni pasovnozaporni filter velikosti 512×512 , $r = 80$, $r_2 = 130$	50
47	Gaussov nizkoprepustni filter velikosti 512×512 , $\sigma = 55$	51
48	Gaussov visokoprepustni filter velikosti 512×512 , $\sigma = 55$	51
49	Gaussov pasovnoprepustni filter velikosti 512×512 , $\sigma = 25$, $\mu = 100$	51
50	Gaussov pasovnozaporni filter velikosti 512×512 , $\sigma = 25$, $\mu = 100$	51
51	Filter z zapornimi črtami velikosti 512×512 , $\alpha_1 = 0^\circ$, $d_1 = 10$, $\alpha_2 = 75^\circ$, $d_2 = 20$, $\sigma = 15$	51
52	Slika Fourierjeve transformacije z dodanim filtrom, ustvarjenim z risanjem, velikosti 512×512	51
53	Diagram poteka odpiranja slikovne datoteke	53
54	Diagram poteka (inverzne) hitre Fourierjeve transformacije slike	53
55	Diagram poteka (inverzne) hitre Fourierjeve transformacije M elementov	54
56	Različna izbira metode svinjenja slike karirastega vzorca	55
57	Intenzitetne transformacije rentgenograma prsnega koša	56
58	Upragovanje slike s tatujem metulja in slike krvnega razmaza	56
59	Povprečenje slike marjetice s konvolucijo	57
60	Laplaceov operator na sliki marjetice (L)	58
61	Prewittov in Sobelov operator na sliki žabe (R)	59
62	Ostrenje z metodo <i>unsharp masking</i> (UM) na sliki Meseca	60
63	Ostrenje z metodo <i>unsharp masking</i> (UM) na rentgenogramu prsnega koša	60
64	Fourierjeve transformacije kroga	61
65	Fourierjevi transformaciji pravokotnikov	61
66	Fourierjevi transformaciji mreže in zvezde	61

67	Idealni nizkopasovni filter na sliki abecede	62
68	Gaussov nizkopasovni filter na sliki abecede	62
69	Idealni viskopasovni filter na sliki abecede	63
70	Centrirani viskopasovni filtri	63
71	Popravilo slike marjetice, poškodovane s sinusnim šumom	64
72	Odstranitev notnega črtovja iz poškodovane slike partiture	65
73	Izvirne barvne fotografije	66

Tabele

1	Primer odvajanja pikslov v eni vrstici	22
2	Nekaj Fourierjevih koeficientov c_n za funkcijo pravokotnega impulza Π	33
3	Bitreverzno zaporedje 8 elementov	46

1 UVOD

„*Today everything exists to end in a photograph.*“
– SUSAN SONTAG, ameriška pisateljica in aktivistka

Digitalne slike nas spremljajo na vsakem koraku – pregledujemo jih na svojih pametnih telefonih, prenašamo jih s spleta, zajemamo jih z digitalnimi fotoaparati. Z razvojem digitalne tehnike pa so nam postala dostopnejša tudi mnoga orodja za obdelavo slik, kot so računalniški programi *Adobe Photoshop*, *Gimp* in *XnView* ali mobilna aplikacija *PicsArt*. Čeravno se programi med seboj razlikujejo po videzu ali številu funkcionalnosti, so njihovi osnovni algoritmi za obdelavo slik enaki.

1.1 Cilji naloge

Z raziskovalno nalogo želim odstretiti matematično podlago, ki nam že od svojih zametkov v 20. stoletju pa vse do danes služi pri obdelavi digitalnih slik. Izmed mnogih podpodročij se bom osredinil na filtriranje v prostorski in frekvenčni domeni ter pri vsakem opisal nekaj osnovnih postopkov obdelave. Poudariti želim matematično izpeljavo filtriranja v frekvenčni domeni (Fourierjeva transformacija), izdelal pa bom tudi računalniški program, v katerem bo možno preizkusiti v nalogi opisane metode.

1.2 Metodologija

S preučevanjem literature o postopkih digitalne obdelave slik sem povzel njihove osnovne lastnosti in se pri tem oziral predvsem na njihovo matematično ozadje, nakar sem s pridobljenim vedenjem sestavil računalniški program s sposobnostmi osnovne obdelave digitalnih slik.

2 TEORETIČNI DEL

2.1 Osnove digitalne obdelave slik

Pojem digitalna obdelava slik zajema kopico različnih metod, procesov in algoritmov, s katerimi operiramo na vizualnih informacijah, predstavljenih v digitalnem zapisu. V splošnem lahko postopke razdelimo v spodnje skupine. (Povzeto po PARKER, 2011)

Zajemanje podatkov Skorajda obvezen postopek za kakršno koli obdelavo je zajemanje slikovnih podatkov, ki običajno poteka z merjenjem količine vpadne svetlobe z merilniki svetlosti in slikovnimi senzorji CCD ter CMOS. Zajete podatke nato s procesom digitalizacije pretvorimo v digitalno sliko.

Filtriranje in izboljšave Med postopke filtriranja in izboljšav sodijo najrazličnejše metode, namenjene izboljšanju vizualne kakovosti slike ali olajšanju nadaljnje obdelave in interpretacije. Splošni postopek ne obstaja, saj vsaka slika zahteva svojevrstno izboljšavo, končni sodnik kakovosti izboljšave pa je človek. Filtriranje se v grobem deli na dve področji, in sicer na filtriranje v prostorski ter frekvenčni domeni, osnovne metode pa so sprememba kontrasta, mehčanje in ostrenje, histogramska obdelava in filtriranje visokih ali nizkih frekvenc.

Restavriranje in rekonstrukcija V nasprotju s filtriranjem lahko restavriranje definiramo in njegovo učinkovitost izmerimo objektivneje, saj je njegov namen povrniti izvirne informacije iz popačene, poškodovane ali uničene slike. Postopki vključujejo predvsem odstranitev neželenega šuma in frekvenc.

Metode rekonstrukcije lahko med drugim najdemo v medicini, kjer si z matematičnimi algoritmi pomagajo pri slikanju z magnetno resonanco (MRI) in računalniški tomografiji (CT). Tako iz več zajetih slik z različnih perspektiv dobijo končno sliko (lahko tudi tridimenzionalno).

Barvna obdelava V področje upravljanja z barvami sodijo teorije različnih barvnih modelov, s katerimi opisujemo in prikazujemo barve, barvni in tonski popravki, odstranjevanje barvnega šuma in barvne transformacije.

Valčne transformacije (ang. *wavelet transform*) Podobne Fourierjevim transformacijam, valčne transformacije predstavljajo slike v drugačnih domenah, in sicer v večresolucijskih piramidah. Metode se razvijajo šele zadnjih nekaj desetletij.

Stiskanje Stiskanje (kompresija) je namenjeno zmanjševanju količine podatkov, ki so potrebni za zapis slike. To je seveda mogoče, saj slike pogosto vsebujejo veliko količino odvečnih podatkov. Področje stiskanja je izjemno pomembno, saj zaradi varčevanja s podatkovnim prostorom skorajda vsako digitalno sliko pred shranjevanjem nekoliko kompresiramo. Obstaja ogromno različnih metod, s katerimi smo pogosto v stiku (npr. slike JPEG) in ki se nadgrajujejo v metode za stiskanje digitalnih videoposnetkov.

Segmentacija in morfološka obdelava S postopki segmentacije in kasneje morfološke obdelave sliko razdelimo v posamezne regije in objekte, prepoznamo položaje črt in vzorcev, opišemo geometrijske oblike ali sliko prikažemo s skeletom. Ti postopki so eni najtežjih v celotni slikovni obdelavi, njihov končni rezultat pa je seznam posameznih sestavnih objektov slike.

Prepoznavanje Prepoznavanje lahko sledi segmentaciji in morfološki obdelavi ter je, kot pove ime, namenjeno prepoznavanju objektov na sliki, to pomeni uvrščanje objektov glede na njihove vizualne značilnosti v vnaprej določene razrede. V praksi to pomeni, da algoritem na sliki s tremi živalmi prepozna, npr. psa, mačko in žirafa. Pomemben element so tudi baze podatkov z deskriptorji posameznih objektov.

Opisovanje Sliko lahko opišemo z različnimi topološkimi in statističnimi lastnostmi (korelacija, kontrast, homogenost, entropija ...).

V svoji raziskovalni nalogi bom na kratko predstavil zajemanje podatkov, glavno področje pa bo predvsem filtriranje (tako v prostorski kot v frekvenčni domeni).

2.1.1 Uporabnost digitalne slikovne obdelave

Od začetka uporabe digitalne obdelave slike – eden prvih takih poskusov je bilo pošiljanje fotografij po čezoceanskem podmorskem kablu med Londonom in New Yorkom, pri katerem je prenos fotografije trajal slabe tri ure – pa do danes so se postopki obdelave slik prenesli v mnogo področij, ki zbirajo informacije iz okolja v različnih spektrih elektromagnetnega valovanja. Naj začnem pri medicini, kjer radiologija z metodami rentgenske, računalniške ali pozitronske tomografije pridobiva vizualne informacije o stanju tkiv v človekovem telesu.

Drugi spekter elektromagnetnega valovanja – ultravijolični – s pridom uporabljajo biokemiki, tako da s fluorescenčno mikroskopijo preiskujejo in opazujejo živa tkiva. V spektru vidne in infrardeče svetlobe se preverja kakovost in popolnost tovarniških izdelkov (iskanje napak v proizvodnem procesu), s satelitskimi slikami lahko

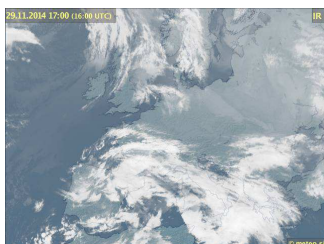
raziskujemo geografske, demografske in meteorološke pojave.

Radijske valove uporabljajo za radarske posnetke tal (saj lahko penetrirajo skozi oblake), podobno se jih uporablja v medicini pri preiskavah z magnetno resonanco. V širokih spektrih valovanja raziskujejo astronomi in astrofiziki, ki z različnimi analizami preučujejo sestavo, starost in oddaljenost nebesnih objektov.

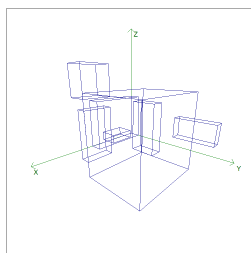
Digitalna obdelava služi tudi državni varnosti in policiji, kjer z metodami prepoznavanja obrazov in prstnih odtisov identificirajo osebe.

Nenazadnje pa je digitalna fotografija postala tudi svoja umetniška smer, ki se razširja zadnjih nekaj desetletij.

Primeri različnih digitalnih slik so zbrani na slikah 1–6.



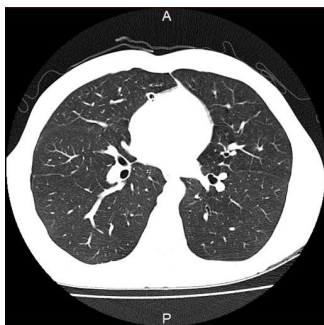
Slika 1: Vremenska slika oblačnosti nad Evropo, posneta v infrardečem spektru (*Vir: ARSO, 2014*)



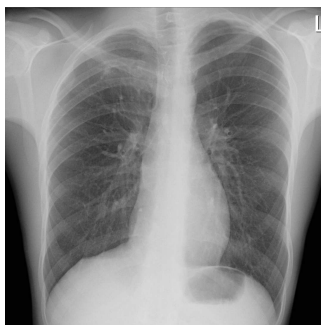
Slika 2: Računalniško generirana slika kvadrov v perspektivni projekciji (*Vir: lasten*)



Slika 3: Galaksija Sombrero (M104), posneta z vesoljskim teleskopom Hubble (*Vir: NASA, 2014*)



Slika 4: Prečna rezina prsnega koša, posneta z računalniško tomografijo (CT) (*Vir: lasten*)



Slika 5: Rentgenski posnetek prsnega koša (*Vir: lasten*)



Slika 6: Endotelijske celice pljučne arterije goveda, posnete s fluorescenčno mikroskopijo (*Vir: KANDASAMY, 2014*)

2.2 Zajemanje in predstavljanje slike

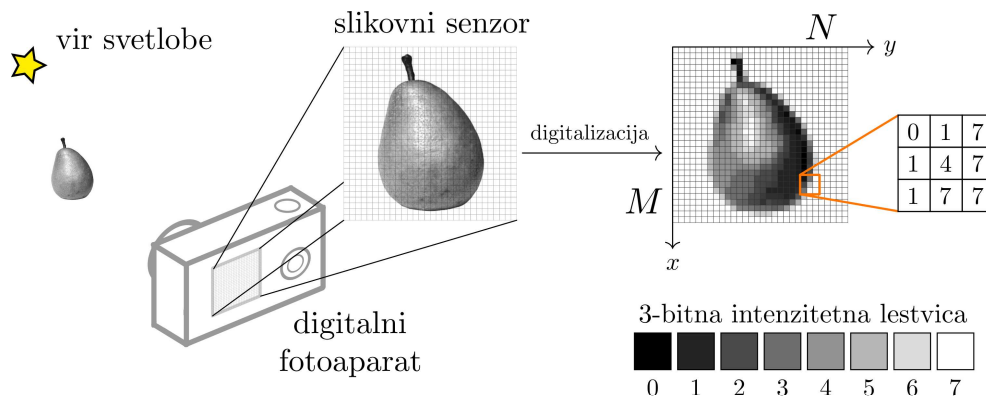
Digitalna slika lahko v glavnem nastane na dva načina: z zajemanjem vizualnih informacij iz okolja ali z računalniškim generiranjem. Osnovni koncept zajemanja informacij iz okolja bom predstavil na primeru digitalnega fotoaparata, a se tudi drugačni načini zajemanja v splošnem držijo sledečega koncepta. V prikazanem postopku bo končni rezultat slika v sivinah, zajemanje in obravnavanje barv bo omenjeno v podpoglavju 2.3.

Vizualne informacije v okolju so predstavljene z **elektromagnetnim valovanjem** različnih valovnih dolžin in jih lahko razdelimo v skupine glede na njihovo valovno dolžino λ (radijski valovi z najdaljšimi valovnimi dolžinami, toplotni IR-valovi, vidna svetloba, ultravijolična svetloba, rentgenski žarki in žarki γ z najkrajšimi valovnimi dolžinami). Glede na potrebe izberemo, katero skupino valov želimo meriti – fotografiranje z digitalnim fotoaparatom zajema elektromagnetno valovanje v vidnem spektru svetlobe (valovna dolžina približno od 390 nm do 700 nm). Objekt fotografiranja osvetlimo z vidno svetlobo, ki se od njega odbije in pripotuje do fotoaparata, kjer je s sistemom leč projicirana na slikovni senzor. Tega sestavlja končno mnogo točkovnih senzorjev, razporejenih v mrežo, ki merijo energijo vpadle svetlobe, kar predstavlja **svetlobno intenziteto**. Intenziteta svetlobe, ki pade na senzor, je odvisna od osvetljenosti objekta in njegove odbojnosti. V našem trenutnem primeru obravnavamo samo celokupno intenziteto svetlobe, torej neodvisno od valovne dolžine. Končna slika zato ne bo barvna, temveč v sivinah. Svetloba iz okolja je analogna količina, a se zaradi končno mnogo točkovnih senzorjev **diskretizira** – temu procesu rečemo vzorčenje (ang. *sampling*). Zadnji korak je **kvantizacija**, kjer vrednosti zajetih intenzitet razdelimo v končno mnogo enako velikih intervalov. Dobljeno digitalno sliko lahko sedaj predstavimo z matriko velikosti $M \times N$, ki ima enake dimenzije kot slikovni senzor. Posamezen element $a_{i,j}$ matrike predstavlja vrednost intenzitete svetlobe po kvantizaciji in ga imenujemo *piksel* ali *slikovna točka*. Matriko slike velikosti $M \times N$ bom v nalogi izražal s funkcijo $\mathcal{S}(x, y)$, pri čemer je $0 \leq x < M$ indeks vrstice in $0 \leq y < N$ indeks stolpca¹:

$$\mathcal{S}(x, y) = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} =$$

¹V praksi se pogosto izraža *ločljivost slike*, tako da se navede podatek *širina*×*višina*; v nalogi bom kot podatek o dimenzijah slike uporabil matematično ustrežnejšo različico, in sicer velikost matrike, tj. *višina*×*širina*.

$$= \begin{bmatrix} \mathcal{S}(0,0) & \mathcal{S}(0,1) & \cdots & \mathcal{S}(0,N-1) \\ \mathcal{S}(1,0) & \mathcal{S}(1,1) & \cdots & \mathcal{S}(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}(M-1,0) & \mathcal{S}(M-1,1) & \cdots & \mathcal{S}(M-1,N-1) \end{bmatrix}$$



Slika 7: Zajemanje digitalne slike z digitalnim fotoaparatom. Slikovni senzor vsebuje 30×30 točkovnih senzorjev, zato je tudi digitalna slika velikosti 30×30 . Svetlobne intenzitete so kvantizirane v 8 svetlostnih intenzitet.

Pri kvantizaciji slike moramo sprejeti pomembno odločitev: število intervalov svetlobne intenzitete. Seveda je povsem očitno, da z več intervali dobimo tudi več informacij o objektu. Zaradi anatomsko-fizioloških omejitev našega očesa² in enote digitalnih informacij – 1 bajt = 8 bit – so danes najbolj razširjene slike z 256 sivinskimi stopnjami.

Zaradi binarnih konstrukcij digitalnih vezij običajno izberemo število intervalov, ki je potenca 2, torej $L = 2^k$. Slika z 2^k diskretnimi stopnjami intenzitete poimenujemo k -bitna slika, saj je za zapis intenzitete enega piksla potrebnih k bitov. Iz tega sledi, da je za zapis celotne slike velikosti $M \times N$ potrebnih kMN bitov.

Slika, predstavljena kot svetlostna intenziteta v dveh razsežnostih $\mathcal{S}(x, y)$ oziroma kot funkcija dveh prostorskih spremenljivk, imenujemo **slika v prostorski domeni**. To ni edina možna predstavitev, saj lahko sliko s transformacijo prestavimo tudi v kako drugo domeno, npr. frekvenčno, kot bomo to storili s Fourierjevimi transformacijami³.

V raziskovalni nalogi bom uporabljal 8-bitne slike z 256 sivinami, razen kjer je

²Delovanje očesa je sicer zelo zapleteno, vendar je v splošnem oko sposobno zaznati približno 20 različnih sivinskih odtenkov (ORLOFF, 1997, str. 483), trenirano oko pa tudi od 50 do 100 (NERI, 2008, str. 57).

³Za potrebe obdelave slik obstaja še mnogo drugih transformacij, kot so kosinusna, sinusna, Houghova, Hadamardova in Haarova (TASIĆ, 2002).

drugače omenjeno. Svetlobne intenzitete bodo torej lahko zavzele diskretne vrednosti na intervalu $[0, 255]$.

2.3 Upravljanje z barvami

Barva je psihofiziološki občutek, ki ga povzroči elektromagnetno valovanje različnih frekvenc znotraj območja vidnega spektra (približno od 390 nm do 700 nm). Človeško mrežnico sestavljajo trije različni tipi čepkov, ki nam omogočajo zaznavanje barv in ki se razlikujejo po lastnih absorpcijskih spektrih. Vrhovi njihovih spektrov so pri valovnih dolžinah 445 nm, 535 nm in 575 nm, kar približno ustreza modri, zeleni in rdeči barvi. Te tri barve imenujemo osnovne barve in z linearnim sestavljanjem različnih deležev posamezne barve lahko dobimo množico drugih barv. (GONZALES, 2008)

Barvna digitalna slika se od sivinske razlikuje v tem, da je vsak njen piksel namesto z eno svetlobno intenziteto zapisan z več svetlobnimi intenzitetami, ki jim pravimo barvni kanali ali barvne komponente. Tako tudi slikovni senzor fotoaparata sestavlja več tipov točkovnih senzorjev, ki so občutljivi samo na določene frekvence. V barvnem modelu *RGB* (*red*, *green*, *blue*), ki ga bom tudi sam uporabljal v nalogi, je vsak piksel zapisan s tremi barvnimi kanali: rdečim, zelenim in modrim. Ker je vsak izmed treh barvnih kanalov 8-biten – intenzitetne stopnje posameznega kanala torej ležijo na diskretnem intervalu $[0, 255]$ –, pravimo, da ima taka slika **24-bitno barvno globino**. Posamezen piksel lahko tako zavzame eno izmed $(2^8)^3 = 16\,777\,216$ različnih barv.

2.3.1 Sivljenje slike⁴

Sliko v barvnem modelu *RGB*, pri kateri je vsak piksel zapisan s komponentami I_R , I_G in I_B , pretvorimo v sivinsko tako, da vsakemu pikslu slike določimo samo eno svetlostno intenziteto I . Določiti moramo torej funkcijo, ki iz danih treh barvnih komponent izračuna samo eno svetlostno intenziteto. To storimo z aritmetično sredino.

Aritmetična sredina kanalov

$$I = \frac{I_R + I_G + I_B}{3}$$

⁴Skovani izraz je v slovenskem prostoru sicer besedna novota, a se mi zdi primerno poimenovanje za proces spreminjanja barvne slike v sivinsko sliko.

Tehtana aritmetična sredina Komponente lahko utežimo z utežmi k_i .

$$I = \frac{k_R I_R + k_G I_G + k_B I_B}{k_I + k_G + k_B}$$

Ena izmed pogosto uporabljenih kombinacij uteži v barvnem modelu *RGB* se imenuje *luminiscenca* (STOKES, 1996):

$$I = 0,2126I_R + 0,7152I_G + 0,0722I_B$$

2.4 Operacije med dvema slikama

Med dvema slikama lahko izvedemo štiri osnovne binarne operacije, in sicer seštevanje, odštevanje, množenje in deljenje. Pri tem je treba poudariti, da operacija množenje ni ekvivalentna matričnemu množenju (po metodi produkta vrstic s stolpci), temveč je tako kot seštevanje in odštevanje operacija med istoležnimi elementi; enako velja za deljenje.

$$\begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix} \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{bmatrix} = \begin{bmatrix} a_{0,0}b_{0,0} & a_{0,1}b_{0,1} \\ a_{1,0}b_{1,0} & a_{1,1}b_{1,1} \end{bmatrix}$$

Pri takšnih operacijah se lahko zgodi, da nova vrednost intenzitete ne leži znotraj danega intervala diskretnih vrednosti od 0 do $L - 1$. Težavo odpravimo tako, da: negativnim vrednostim pripišemo vrednost 0; vrednostim, višjim od najvišje možne, pripišemo $L - 1$; morebitna necela števila pa zaokrožimo na najbližje celo število. Opisano funkcijo bom v raziskovalni nalogi označeval z ξ :

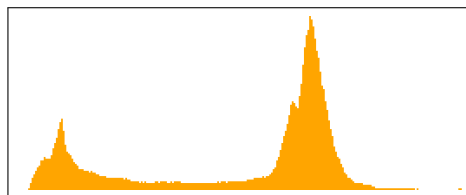
$$\xi(x) = \begin{cases} 0 & ; x \leq 0 \\ L - 1 & ; x \geq L - 1 \\ \text{zaokroži}(x) & ; x > 0 \wedge x < L - 1 \end{cases} \quad (2.1)$$

2.5 Predstavitev s histogramom

Histogram slike predstavlja relativno frekvenco pojavljanja posameznih svetlostnih intenzitet. Na vodoravno os naneseemo vse možne intenzitete (vrednosti, ki jih lahko zavzame posamezen piksel), na navpično os pa naneseemo število pikselov s posamezno intenziteto. Že bežen pogled na histogram nam lahko razkrije splošne lastnosti o sliki. Histogram temne slike bo pomaknjen močno v levo, medtem ko bodo vrednosti na histogramu svetle slike prevladovale na desni polovici. Vrednosti slike z visokim kontrastom bodo imele visoko razpršenost (pogosto bosta opazna tudi dva vrhova – en v levi in drugi v desni polovici histograma), medtem ko bo histogram slike z nizkim kontrastom imel zgoščene vrednosti.



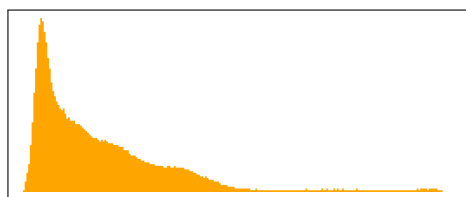
Slika 8: Slika žabe



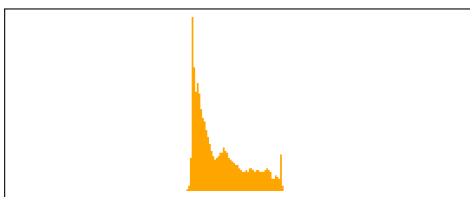
Slika 9: Histogram slike žabe



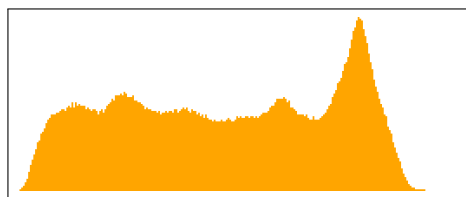
Slika 10: Histogram svetle slike



Slika 11: Histogram temne slike



Slika 12: Histogram slike z nizkim kontrastom



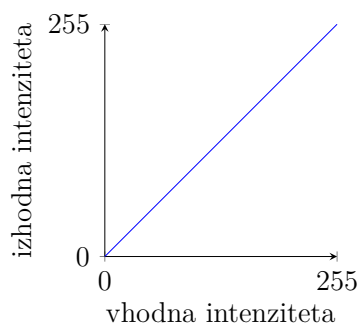
Slika 13: Histogram slike z visokim kontrastom

2.6 Intenzitetne transformacije

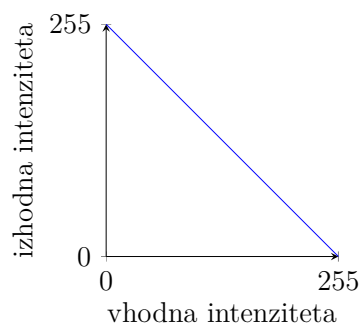
Zamislimo si, da želimo sliko posvetliti – vrednosti na histogramu bi torej morali zamakniti v desno, kar pomeni, da bi vsakemu pikslu slike prišteli neko konstantno vrednost k . Podobno bi sliko potemnilo tako, da bi od vsakega piksla vrednost odšteli.

$$\mathcal{S}'(x, y) = \xi(\mathcal{S}(x, y) \pm k) \quad ; \quad k \in \mathbb{N}_0$$

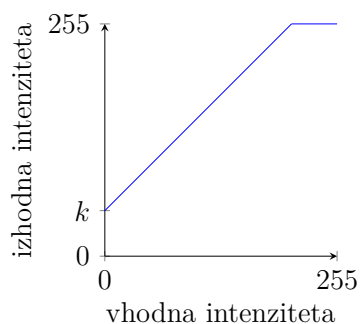
Take preslikave, pri katerih novo intenziteto podamo kot funkcijo v odvisnosti od prvotne intenzitete, imenujemo **intenzitetne transformacije** in jih najlažje predstavimo z njihovimi grafi. Tako je na prvem grafu prikazana identična transformacija, ki slike ne spremeni, na drugem je prikazana transformacija za negativ slike, nadalje sta prikazana grafa svetljenja in temnenja.



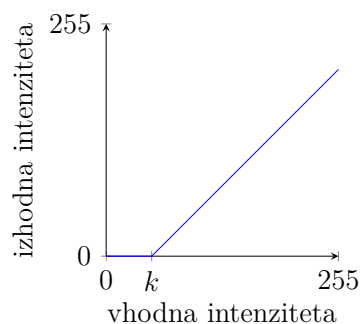
Slika 14: Identična transformacija



Slika 15: Transformacija negativna



Slika 16: Transformacija svetljenja



Slika 17: Transformacija temnenja

Kontrast slike povečamo tako, da histogram raztegnemo, in zmanjšamo tako, da histogram skrčimo. Prva odločitev je izbira intenzitetne vrednosti $0 \leq \mu \leq 255$, ki bo služila kot središče raztega, nato pa izberemo še koeficient raztega $\chi \geq 0$. Transformacijo raztega kontrasta sem sedaj izvedel tako, da graf identične transformacije premaknem navpično navzdol za μ , nato ga raztegnem (oz. skrčim) v navpični smeri s faktorjem χ in ponovno premaknem za μ navzgor:

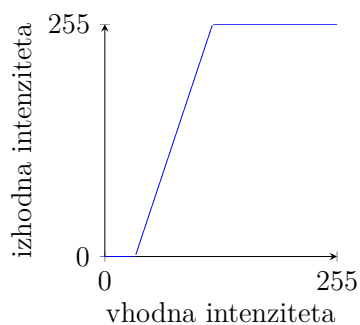
$$\mathcal{S}'(x, y) = \xi((\mathcal{S}(x, y) - \mu)\chi + \mu)$$

V praksi se pogosteje uporabljajo sigmoidne funkcije (njihov graf ima obliko črke „S“), ki imajo bolj „gladko“ obliko (graf seveda ni zvezen, saj so svetlostne intenzitete diskretne vrednosti).

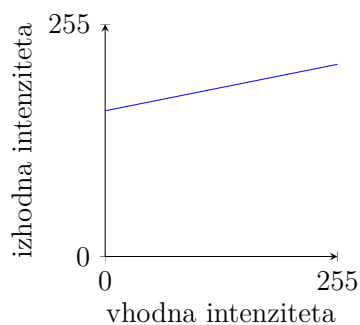
Pomembna točkovna operacija je tudi **upragovanje** (ang. *thresholding*), pri kateri določimo mejno intenzitetno vrednost μ in izvedemo transformacijo:

$$\mathcal{S}'(x, y) = \begin{cases} 255 & ; \mathcal{S}(x, y) \geq \mu \\ 0 & ; \mathcal{S}(x, y) < \mu \end{cases}$$

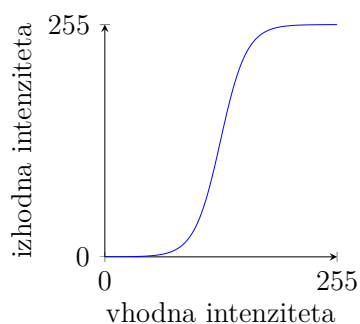
Vsi piksli z intenziteto nižjo od μ bodo postali črni, ostali bodo postali beli.



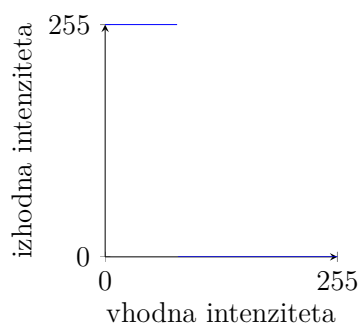
Slika 18: Transformacija raztega kontrasta, $\mu = 50$ in $\chi = 2$



Slika 19: Transformacija raztega kontrasta, $\mu = 200$ in $\chi = 0,8$



Slika 20: Transformacija raztega kontrasta s sigmoidno funkcijo



Slika 21: Transformacija uprakovanja

2.7 Obdelava v prostorski domeni

2.7.1 Konvolucija

Konvolucija je matematična operacija med dvema funkcijama, ki je definirana kot

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(t)g(x - t) dt$$

Za nas bo pomembna diskretna konvolucija med dvema matrikama, in sicer med matriko naše digitalne slike velikosti $M \times N$ in drugo matriko, ki jo imenujemo **konvolucijsko jedro** (ang. *convolution kernel*) velikosti $m \times n$.

$$\mathcal{K}(x, y) * \mathcal{S}(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b \mathcal{K}(s, t)\mathcal{S}(x - s, y - t)$$

pri čemer $a = \frac{m-1}{2}$ in $b = \frac{n-1}{2}$. Konvolucijo je najlažje prikazati grafično. Treba je povedati, da bo konvolucijsko jedro v moji raziskovalni nalogi zmerom kvadratna

matrika z lihim številom stolpcev in vrstic. Prav tako njene indekse označimo nekoliko drugače, in sicer je središče $(0,0)$ v središčnem elementu matrike, ne levo-zgornjem kot običajno.

Konvolucijo izvedemo tako, da konvolucijsko jedro najprej zrcalimo prek obeh osi, tj. zavrtimo za 180° ⁵. Sedaj na matriko slike postavimo konvolucijsko jedro tako, da se njegov središčni element pokriva s prvim (levo-zgornjim) elementom matrike slike. Seštejemo vse istoležne produkte in dobljeni rezultat predstavlja prvi element v matriki rezultata konvolucije. Konvolucijsko jedro premaknemo za en element v desno in ponovno izračunamo vsoto produktov istoležnih elementov. Postopek ponovimo za vse elemente matrike slike. Kot vidimo, se na robovih matrike slike pojavijo manjkajoči elementi. Glede na potrebe lahko to težavo rešimo različno; jaz bom uporabljal *ničelno zapolnjevanje*, pri katerem manjkajoče elemente nadomestimo z intenziteto 0.

Iz danega opisa postopka je jasno, da pri obdelavi slike s konvolucijo na posamezen piksel končne slike vplivajo njegovi sosednji piksli. Večje kot je konvolucijsko jedro, več sosednjih pikslov je zajetih in vplivajo na končno vrednost piksla. Poiščimo nekaj trivialnih in manj trivialnih konvolucijskih jeder, ki jih bom označeval s \mathcal{K} in opisnim indeksom⁶. Očitno je jedro identitete, ki je lahko poljubno veliko, ima v središču element 1, drugje 0:

$$\mathcal{K}_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

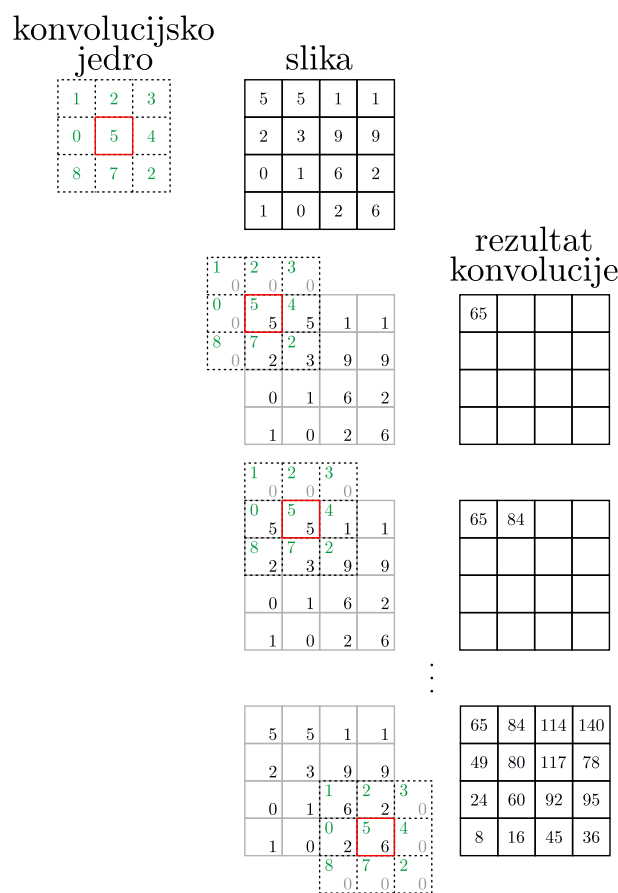
Rezultat konvolucije bo torej identična slika. Če enico v matriki prestavimo na kakšno drugo mesto in ohranimo ničle, dobimo vzporedni premik slike. Namesto enice v sredini lahko uporabimo katero drugo število, ki bo povzročilo spremembo svetlosti vseh pikslov za neki faktor. Tako bo na primer jedro:

$$\mathcal{K}_- = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

celotno sliko potemnilo tako, da bodo piksli imeli polovico svoje prvotne svetlosti. Podobno lahko sliko posvetlimo, tako da vstavimo število večje od 1, na primer:

⁵Če tega ne storimo, se nadaljnji postopek imenuje korelacija.

⁶Te oznake sicer niso standardne in jih v nalogi uporabljam le zaradi ustreznega razločevanja med posameznimi jedri.



Slika 22: Konvolucija digitalne slike

$$\mathcal{K}_+ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Seveda bodo pri množenju s številom večjim od ena nove vrednosti pikslov lahko presegle najvišjo možno vrednost svetlobne intenzitete, zato moramo po konvoluciji vse pikse spustiti skozi funkcijo ξ (glej formulo 2.1).

Pri do sedaj omenjenih konvolucijskih jedrih smo spreminjali samo središčno vrednost, vse ostale so bile 0, kar pomeni, da na končno vrednost piksla vpliva samo začetna vrednost piksla. Sedaj bomo vključili tudi sosednje piksele. Prva zanimiva možnost je povprečenje vrednosti sosednjih pikslov. Nova vrednost piksla bo tako aritmetična sredina sosednjih pikslov. Z velikostjo matrike jedra vplivamo na število pikslov, ki so vključeni v povprečje. Primer za povprečje 8 sosednjih elementov (skupaj jih je torej 9):

$$\mathcal{K}_{\bar{9}} = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Rezultat konvolucije s takim jedrom je „zamegljena“ oziroma omehčana slika, pri kateri so zabrisane ostre meje med objekti.

Navadno povprečenje lahko nadgradimo z utežmi.

$$\mathcal{K}_{\bar{U}} = \frac{1}{13} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Najpogosteje uporabljeno uteženo povprečenje je konvolucija z **Gaussovimi jedrom**, ki je lahko poljubno veliko in se generira z diskretiziranjem Gaussove porazdelitve. Primer takega jedra velikosti 5×5 :

$$\mathcal{K}_{\bar{G5}} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Laplaceov operator Z zgornjimi konvolucijskimi jedri smo sliko zameglili (zmehčali), uporabno pa bi bilo, da bi lahko meglene slike tudi izostrili. Z besedami digitalne slike to pomeni poudariti tista območja slike, kjer so „hitre“ in velike spremembe v svetlosti – taka območja predstavljajo robove objektov. Spremembo svetlosti lahko izrazimo z odvodom. Najprej pogledjmo samo piksele v eni vrstici – funkcija $f(x)$ naj predstavlja posamezne piksele ene vrstice.

$$\frac{df}{dx} = f(x+1) - f(x)$$

f	...	4	4	4	4	5	6	7	7	7	2	2	2	2	...
$\frac{df}{dx}$...	0	0	0	1	1	1	0	0	-5	0	0	0	0	...
$\frac{d^2f}{dx^2}$...	0	0	1	0	0	-1	0	-5	5	0	0	...	

Tabela 1: Primer odvajanja pikselov v eni vrstici

Odvod nekega piksla je torej sprememba svetlosti med danim in naslednjim pikslom. Nadalje poiščemo še drugi odvod, tako da izračunamo razliko med prvim odvodom danega piksla (x) in prvim odvodom prejšnjega piksla ($x - 1$):

$$\begin{aligned}\frac{d^2f}{dx^2} &= f(x+1) - f(x) - (f(x) - f(x-1)) = \\ &= f(x+1) + f(x-1) - 2f(x)\end{aligned}$$

Izraz nadgradimo v dve dimenziji, tako da izrazimo parcialne odvode slike:

$$\begin{aligned}\frac{\partial \mathcal{S}}{\partial x} + \frac{\partial \mathcal{S}}{\partial y} &= (\mathcal{S}(x+1, y) - \mathcal{S}(x, y)) + (\mathcal{S}(x, y+1) - \mathcal{S}(x, y)) = \\ &= \mathcal{S}(x+1, y) + \mathcal{S}(x, y+1) - 2\mathcal{S}(x, y)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{S}}{\partial x^2} + \frac{\partial^2 \mathcal{S}}{\partial y^2} &= \mathcal{S}(x+1, y) + \mathcal{S}(x-1, y) - 2\mathcal{S}(x, y) + \\ &\quad + \mathcal{S}(x, y+1) + \mathcal{S}(x, y-1) - 2\mathcal{S}(x, y) = \\ &= \mathcal{S}(x+1, y) + \mathcal{S}(x-1, y) + \mathcal{S}(x, y+1) + \mathcal{S}(x, y-1) - 4\mathcal{S}(x, y)\end{aligned}$$

Dobljeni izraz imenujemo **Laplaceov operator**:

$$\nabla^2 \mathcal{S}(x, y) = \frac{\partial^2 \mathcal{S}}{\partial x^2} + \frac{\partial^2 \mathcal{S}}{\partial y^2}$$

Celotno sliko bomo torej odvajali tako, da bomo uporabili konvolucijo z jedrom:

$$\mathcal{K}_{\nabla^2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Ker bodo vrednosti konvolucije tudi negativne vrednosti, je smiselno vse vrednosti povečati za 127, ki se nahaja na sredini intervala možnih svetlostnih intenzitet $[0, 255]$ – to bom imenoval *centriranje na sivino*. S tem popravkom ne izgubimo negativnih vrednosti, ki bi jih funkcija ξ preslikala v vrednost 0.

Še nekaj zanimivih konvolucijskih jeder, uporabnih za zaznavanje robov. Oba operatorja imata dve konvolucijski jedri; eno zaznava vodoravne, drugo navpične robove na sliki. V primerih obdelave na koncu naloge sta operatorja omenjena kot xy , kar pomeni, da sem sliko najprej obdelal z operatorjem za vodoravne robove, nato z operatorjem za navpične robove in dobljeni sliki seštel.

Prewittov operator

$$\mathcal{K}_{P_X} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathcal{K}_{P_Y} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobelov operator

$$\mathcal{K}_{S_X} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \mathcal{K}_{S_Y} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2.7.2 Maskiranje

Postopek, pri katerem k prvotni sliki \mathcal{S} prištejem ali odštejem drugo sliko (masko) \mathcal{S}_{maska} in pri tem dobim novo, obdelano sliko \mathcal{S}' , imenujem *maskiranje*⁷:

$$\mathcal{S}' = \mathcal{S} \pm c \cdot \mathcal{S}_{maska} \quad ; \quad c \in \mathbb{R}^+$$

pri tem je c poljubna konstanta, ki določa delež vpliva maske.

Metodo maskiranja uporabljamo predvsem za ostrenje slike. Pri konvoluciji slike s konvolucijskim jedrom Laplaceovega operatorja smo dobili sliko, ki prikazuje položaje ostrih robov na sliki. Če to sliko (masko) odštejemo od izvirne slike, dobimo ostrejšo izvirno sliko.

$$\mathcal{S}' = \mathcal{S} - c \cdot \underbrace{(\mathcal{K}_{\nabla^2} * \mathcal{S})}_{\mathcal{S}_{maska}}$$

Druga metoda ostrenja slik z uporabo maske se imenuje *unsharp masking*⁸. Pri tej metodi dobimo masko slike tako, da izvirno sliko zameglimo (npr. z Gaussovimi konvolucijskim jedrom), nato zamegljeno sliko odštejemo od izvirne slike.

$$\mathcal{S}' = \mathcal{S} + c \cdot \underbrace{(\mathcal{S} - (\mathcal{K}_{G^5} * \mathcal{S}))}_{\mathcal{S}_{maska}}$$

2.8 Predstavitev v frekvenčni domeni

Do sedaj smo sliko obravnavali v prostorski domeni, torej kot funkcijo dveh ravninskih spremenljivk x in y . Sedaj se bomo podali na trnovo pot transformacije slike v frekvenčno domeno.

2.8.1 Fourierjeva vrsta

Joseph Fourier (1768–1830) je ugotovil, da lahko vsako odsekoma zvezno periodično funkcijo zapišemo kot neskončno trigonometrično vrsto. (GONZALES, 2008) Preden se podamo v podrobnejšo razlago, najprej preglejmo osnove.

⁷Zaradi jasnosti sem v zapisu izpustil parametre funkcij; seveda so vse omenjene funkcije – \mathcal{S} , \mathcal{S}' , \mathcal{S}_{maska} , in \mathcal{K} – funkcije dveh prostorskih spremenljivk x in y .

⁸Približen pomenski prevod bi se glasil: *metoda ostrenja z neostro masko*.

Polarni zapis kompleksnega števila Kompleksno število

$$z = a + bi \quad ; \quad a, b \in \mathbb{R}$$

lahko zapišemo tudi v polarni obliki:

$$z = r(\cos \phi + i \sin \phi) \quad ; \quad r \geq 0, \quad 0 \leq \phi < 2\pi$$

pri čemer sta $r = \sqrt{a^2 + b^2}$ in $\phi = \arctan\left(\frac{b}{a}\right)$. Konstanta i je imaginarna enota, za katero velja $i^2 = -1$.

Z Eulerjevo formulo:

$$e^{ix} = \cos x + i \sin x \quad (2.2)$$

lahko kompleksno število izrazimo tudi kot:

$$z = r e^{i\phi}$$

Konjugirano število števila z :

$$\bar{z} = r e^{-i\phi}$$

Trigonometrične funkcije Najprepoznavnejša lastnost funkcij sinus in kosinus je njuna **periodičnost** z osnovno periodo $p = 2\pi$, kar lahko zapišemo kot:

$$\sin(x + 2\pi k) = \sin(x) \quad ; \quad k \in \mathbb{Z}$$

$$\cos(x + 2\pi k) = \cos(x) \quad ; \quad k \in \mathbb{Z}$$

Funkcijama lahko njuno periodo tudi spremenimo, tako da argument x pomnožimo s koeficientom n , kar pomeni, da funkcijo središčno raztegnemo v smeri osi x . Nova osnovna perioda funkcij $\sin(nx)$ in $\cos(nx)$ je $p = \frac{2\pi}{n}$.

Naslednji lastnosti sta **lihost** sinusa in **sodost** kosinusa.

$$\sin(-x) = -\sin(x) \quad ; \quad \forall x \in \mathbb{R}$$

$$\cos(-x) = \cos(x) \quad ; \quad \forall x \in \mathbb{R}$$

Zgornje lastnosti nam dajo posebne lastnosti določenega integrala, in sicer:

$$\int_{-n\pi}^{n\pi} \sin(x) \, dx = 0 \quad ; \quad \forall n \in \mathbb{Z}$$

$$\int_{-n\pi}^{n\pi} \cos(x) \, dx = 0 \quad ; \quad \forall n \in \mathbb{Z}$$

Potrebovali bomo še naslednja nedoločena integrala

$$\int \sin(nx) \, dx = -\frac{1}{n} \cos(nx) + C$$

$$\int \cos(nx) \, dx = \frac{1}{n} \sin(nx) + C$$

Definicija 1 Naj bo f zvezna funkcija na intervalu $[a, b]$. Vrsti

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx)$$

v kateri so koeficienti

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \, dx$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) \, dx \quad ; \quad n \in \mathbb{N}$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) \, dx \quad ; \quad n \in \mathbb{N}$$

pravimo **Fourierjeva vrsta** funkcije f .

Brez dokaza privzemimo še naslednji izrek:

Izrek 1 Če je periodična funkcija f z osnovno periodo 2π odsekoma zvezna na $[-\pi, \pi]$ in ima v vsaki točki intervala levo in desno limito, je njena Fourierjeva vrsta **konvergentna**. Vrednost vrste je enaka $f(x)$ v vsaki točki, kjer je f zvezna.

Za začetek bom spodnjo funkcijo, ki predstavlja pravokotni signal, izrazil s Fourierjevo vrsto.

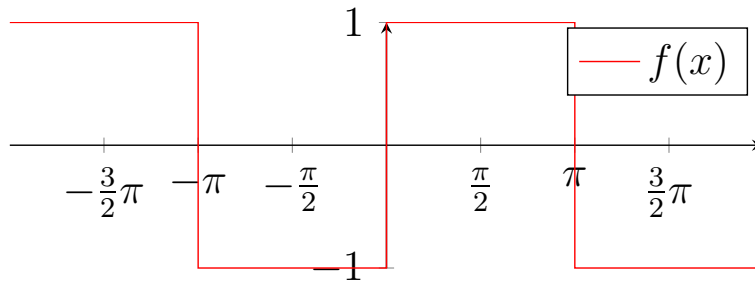
$$f(x) = \begin{cases} -1 & ; \quad -\pi < x \leq 0 \\ 1 & ; \quad 0 < x \leq \pi \end{cases}$$

$$f(x + 2\pi) = f(x) \quad ; \quad \forall x \in \mathbb{R}$$

Funkcija je torej periodična z osnovno periodo 2π in tudi odsekoma zvezna. Za lažjo predstavitev je na sliki 23 tudi njen graf⁹.

Poiščimo sedaj koeficiente Fourierjeve vrste. Očitno je, da velja:

⁹Na tem in nekaterih kasnejših grah so vrisane tudi navpične črte, ki sicer niso del grafa funkcije, temveč so prikazane zgolj zaradi poudarjanja pravokotne oblike signala.



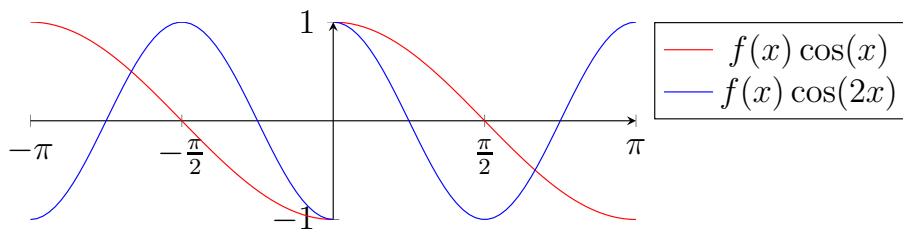
Slika 23: Graf funkcije pravokotnega signala

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx = 0$$

Naslednji koeficient je a_n :

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) dx$$

Če pogledamo graf funkcije produkta $f(x)$ in $\cos(x)$, vidimo, da je integral na $[-\pi, \pi]$ enak 0.



Slika 24: Graf funkcij $f(x)\cos(x)$ (rdeče) in $f(x)\cos(2x)$ (modro)

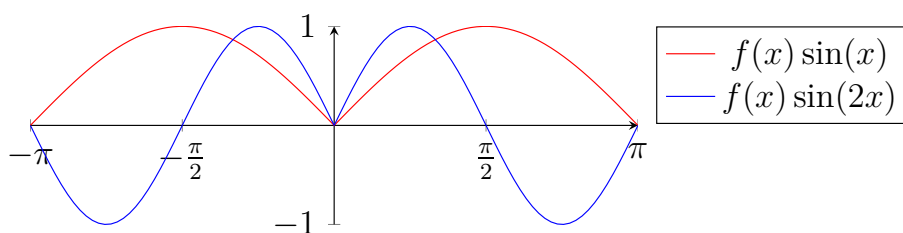
Enako velja tudi za vse ostale $n \in \mathbb{N}$, ki sicer povzročijo manjšanje osnovne periode kosinusa, a je integral na $[-\pi, \pi]$ še zmerom enak 0, torej:

$$a_n = 0$$

Izračunajmo še b_n .

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) dx$$

Poglejmo, kaj se dogaja s tem integralom pri različnih n .



Slika 25: Graf funkcij $f(x) \sin(x)$ (rdeče) in $f(x) \sin(2x)$ (modro)

$$n = 1$$

$$\int_{-\pi}^{\pi} f(x) \sin(x) dx = 2 \int_0^{\pi} \sin(x) dx = 2(-\cos(x)|_0^{\pi}) = 2(-(-1) + 1) = 4$$

$$n = 2$$

$$\int_{-\pi}^{\pi} f(x) \sin(2x) dx = 2 \int_0^{\pi} \sin(2x) dx = 2 \cdot \left(-\frac{1}{2}(\cos(2x)|_0^{\pi})\right) = -1 \cdot (1 - 1) = 0$$

$$n = 3$$

$$\int_{-\pi}^{\pi} f(x) \sin(3x) dx = 2 \int_0^{\pi} \sin(3x) dx = 2 \cdot \left(-\frac{1}{3}(\cos(3x)|_0^{\pi})\right) = -\frac{2}{3}(-1 - 1) = \frac{4}{3}$$

Vidimo, da je za vsak n odločilen faktor v integralu $\cos(n\pi)$, ki zavzame vrednosti -1 pri lihih n in 1 pri sodih n . Iz tega sklepamo, da je vrednost integrala pri sodih n enaka 0 , pri lihih n pa:

$$\int_{-\pi}^{\pi} f(x) \sin(nx) dx = -\frac{2}{n} \cdot (-2) = \frac{4}{n}$$

Iz tega sledi:

$$b_n = \frac{1}{\pi} \cdot \frac{4}{n} = \frac{4}{n\pi}$$

Ker imajo členi na sodih mestih vrednost 0 , se jih bomo elegantno znebili in ostanejo nam le še členi na lihih mestih:

$$b_n = \frac{4}{(2n-1)\pi}$$

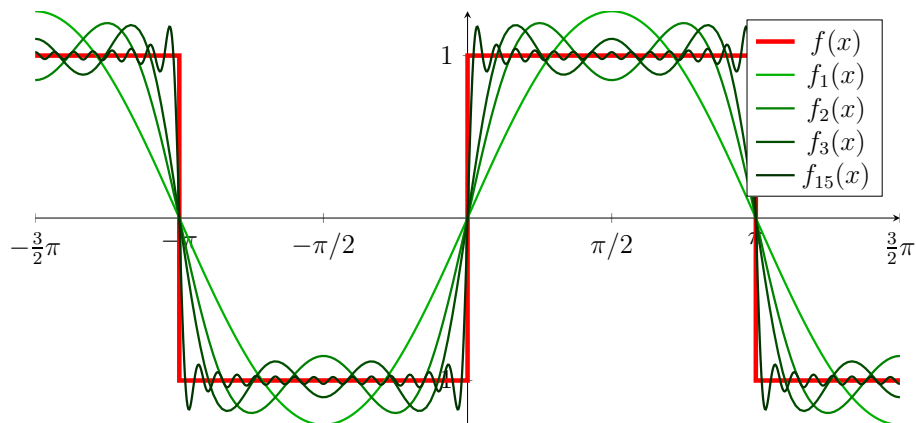
Končna Fourierjeva vrsta se tako glasi:

$$f(x) = \sum_{n=1}^{\infty} \frac{4}{(2n-1)\pi} \sin(nx)$$

ali če izpišemo prvih nekaj členov:

$$f(x) = \frac{4}{\pi} \sin(x) + \frac{4}{3\pi} \sin(3x) + \frac{4}{5\pi} \sin(5x) + \frac{4}{7\pi} \sin(7x) + \dots$$

Zares, na grafu lahko vidimo, da s povečevanjem števila členov vrsta konvergira k prvotni funkciji $f(x)$.



Slika 26: Graf pravokotnega signala $f(x)$ in Fourierjeve vrste $f_i(x)$ s prvimi i členi vrste

Razširitev Fourierjeve vrste Našo Fourierjevo vrsto bomo sedaj razširili na funkcije s poljubno osnovno periodo $p = 2L$.

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{L}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right)$$

Hkrati se spremenijo tudi koeficienti vrste

$$a_0 = \frac{1}{2L} \int_{-L}^L f(x) dx$$

$$a_n = \frac{1}{L} \int_{-L}^L f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

$$b_n = \frac{1}{L} \int_{-L}^L f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$

Zaradi krajšega zapisa definiramo nov simbol:

$$\omega_n = \frac{n\pi}{L}$$

Hkrati bomo zapis s kotnima funkcijama raje spremenili v zapis kompleksnega števila s pomočjo kompleksne eksponentne funkcije. Iz enačbe 2.2 lahko izpeljemo:

$$\sin(\omega_n x) = \frac{1}{2i}(e^{i\omega_n x} - e^{-i\omega_n x})$$

$$\cos(\omega_n x) = \frac{1}{2}(e^{i\omega_n x} + e^{-i\omega_n x})$$

Sedaj poskusimo Fourierjev koeficient a_n preoblikovati.

$$\begin{aligned} a_n &= \frac{1}{L} \int_{-L}^L f(x) \cos(\omega_n x) dx = \frac{1}{L} \int_{-L}^L f(x) \frac{1}{2}(e^{i\omega_n x} + e^{-i\omega_n x}) dx \\ &= \frac{1}{2L} \int_{-L}^L f(x) e^{i\omega_n x} dx + \frac{1}{2L} \int_{-L}^L f(x) e^{-i\omega_n x} dx \end{aligned}$$

Definirali bomo nov Fourierjev koeficient c_n :

$$c_n = \frac{1}{2L} \int_{-L}^L f(x) e^{-i\omega_n x} dx$$

Fourierjev koeficient a_n lahko sedaj zapišemo:

$$a_n = c_n + c_{-n}$$

Naredimo podobno še za koeficient b_n :

$$\begin{aligned} b_n &= \frac{1}{L} \int_{-L}^L f(x) \sin(\omega_n x) dx = \frac{1}{L} \int_{-L}^L f(x) \frac{1}{2i}(e^{i\omega_n x} - e^{-i\omega_n x}) dx = \\ &= \frac{1}{2Li} \int_{-L}^L f(x) e^{i\omega_n x} dx - \frac{1}{2Li} \int_{-L}^L f(x) e^{-i\omega_n x} dx = \\ &= i(c_n - c_{-n}) \end{aligned}$$

Če sedaj pogledamo še koeficient a_0

$$a_0 = \frac{1}{2L} \int_{-L}^L f(x) dx$$

Integrand pomnožimo z $1 = e^0$ in vidimo, da je dobljen izraz enak c_n pri $n = 0$

$$a_0 = \frac{1}{2L} \int_{-L}^L f(x) e^0 dx = c_0$$

Sedaj celotno vrsto zapišimo z novimi koeficienti:

$$\begin{aligned} f(x) &= a_0 + \sum_{n=1}^{\infty} a_n \cos(\omega_n x) + \sum_{n=1}^{\infty} b_n \sin(\omega_n x) = \\ &= c_0 + \sum_{n=1}^{\infty} (c_n + c_{-n}) \cos(\omega_n x) + \sum_{n=1}^{\infty} i(c_n - c_{-n}) \sin(\omega_n x) = \\ &= c_0 + \sum_{n=1}^{\infty} \left((c_n + c_{-n}) \cdot \frac{1}{2}(e^{i\omega_n x} + e^{-i\omega_n x}) + i(c_n - c_{-n}) \cdot \frac{1}{2i}(e^{i\omega_n x} - e^{-i\omega_n x}) \right) = \\ &= c_0 + \sum_{n=1}^{\infty} \frac{1}{2} (c_n e^{i\omega_n x} + c_n e^{-i\omega_n x} + c_{-n} e^{i\omega_n x} + c_{-n} e^{-i\omega_n x} + \\ &\quad + c_n e^{i\omega_n x} - c_n e^{-i\omega_n x} - c_{-n} e^{i\omega_n x} + c_{-n} e^{-i\omega_n x}) = \\ &= c_0 + \sum_{n=1}^{\infty} (c_n e^{i\omega_n x} + c_{-n} e^{-i\omega_n x}) \end{aligned}$$

Dobljeni izraz lahko elegantno poenostavimo tako, da namesto vrste z dvema členoma teče n od $-\infty$ do ∞ . S tem je vključen tudi člen c_0 :

$$\boxed{f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n x}} \quad (2.3)$$

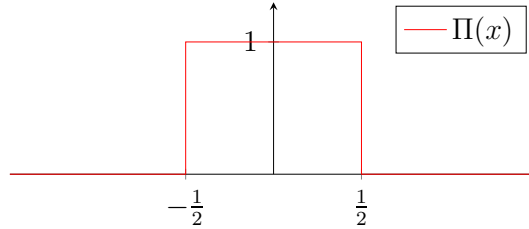
Dobili smo končni strnjeni zapis Fourierjeve vrste s kompleksno eksponentno funkcijo.

2.8.2 Fourierjeva transformacija

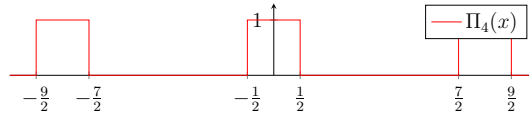
Do sedaj smo se ukvarjali samo s periodičnimi funkcijami z osnovno periodo $p = 2L$. Fourierjeve vrste pa lahko razširimo tudi na neperiodične funkcije, tako da periodo povečujemo prek vseh meja. Našo Fourierjevo vrsto bomo za periodične funkcije pretvorili v **Fourierjevo transformacijo** za neperiodične funkcije.

Naredimo Fourierjevo vrsto za en pravokotni impulz:

$$\Pi(x) = \begin{cases} 1 & ; |x| < \frac{1}{2} \\ 0 & ; |x| \geq \frac{1}{2} \end{cases}$$



Slika 27: Graf pravokotnega impulza



Slika 28: Graf periodičnega pravokotnega impulza, $p = 4$

Ta funkcija seveda ni periodična, zato je ne moremo zapisati s Fourierjevo vrsto. Lahko pa jo zelo enostavno naredimo periodično z osnovno periodo p , tako da k predpisu funkcije dodamo:

$$\Pi(x + kp) = \Pi(x) \quad ; \quad p \in \mathbb{R}, k \in \mathbb{Z}$$

Zaradi težavnosti integracije kompleksne funkcije sem za izračun Fourierjevih koeficientov c_n uporabil spletno matematično orodje WolframAlpha¹⁰.

$$c_n = \frac{1}{2L} \int_{-L}^L \Pi(x) e^{-i\omega_n x} dx = \frac{1}{2} \int_{-1}^1 \Pi(x) e^{-i\omega_n x} dx$$

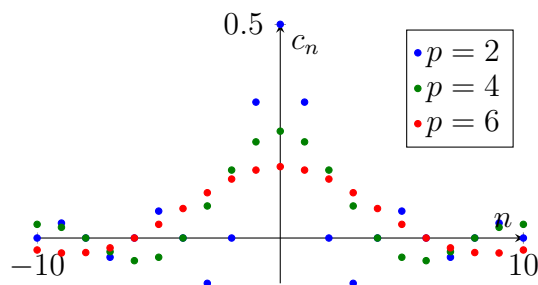
Ker je Π zunaj intervala $[-\frac{1}{2}, \frac{1}{2}]$ enak 0, lahko območje integriranja zmanjšamo:

$$c_n = \frac{1}{2L} \int_{-\frac{1}{2}}^{\frac{1}{2}} \Pi(x) e^{-i\omega_n x} dx$$

Izračunal sem koeficiente c_n za osnovne periode 2, 4 in 6. Nekaj vrednosti je prikazanih v tabeli 2, več jih je prikazanih na grafu 29.

$p \backslash n$	-3	-2	-1	0	1	2	3
2	-0,106	0	0,318	0,5	0,318	0	-0,106
4	0,075	0,159	0,225	0,25	0,225	0,159	0,075
6	0,106	0,138	0,159	0,167	0,159	0,138	0,106

Tabela 2: Nekaj Fourierjevih koeficientov c_n za funkcijo periodičnega pravokotnega impulza Π pri treh različnih periodah. Vrednosti so zaokrožene.



Slika 29: Izračunani Fourierjevi koeficienti c_n periodičnega pravokotnega impulza z osnovnimi periodami 2, 4 in 6.

Vidimo, da koeficienti c_n s povečevanjem osnovne periode pravokotnega impulza vse hitreje konvergirajo k 0, zato graf raztegnemo v smeri ordinatne osi s faktorjem p . Zatem ga še skrčimo v smeri abscisne osi s faktorjem p , in to dvakrat skalirano funkcijo bom imenoval $c' \left(\frac{n}{p} \right)$.

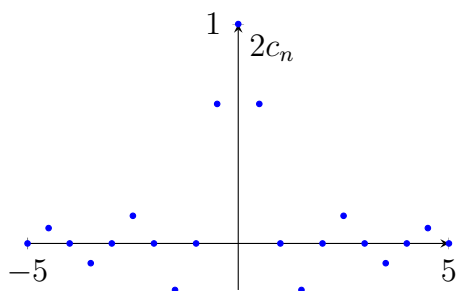
Izkaže se, da začnejo izračunani koeficienti neodvisno od izbrane osnovne periode zavzemati podobno obliko na grafu, zaradi vodoravnega skrčenja pa ležijo točke na grafu pri povečevanju osnovne periode p vedno bolj skupaj. Če torej p povečujemo prek vseh meja, dobimo zvezno funkcijo zvezne spremenljivke (uporabil bom oznako μ), s tem pa smo tudi našo prvotno funkcijo Π ponovno naredili neperiodično.

$$c' \left(\frac{n}{p} \right) = p \cdot \frac{1}{2L} \int_{-L}^L \Pi(x) e^{-i\omega_n x} dx \xrightarrow{L \rightarrow \infty} \hat{\Pi}(\mu) = \int_{-\infty}^{\infty} \Pi(x) e^{-i2\pi x \mu} dx \quad \begin{array}{l} p = 2L \\ \omega_n = \frac{n\pi}{L} \end{array}$$

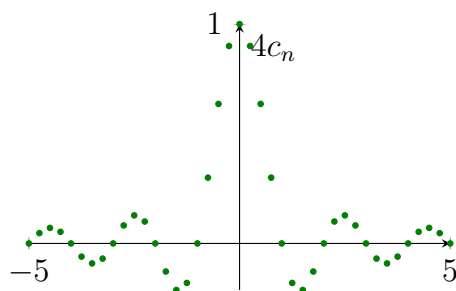
Dobili smo **Fourierjevo transformacijo neperiodične funkcije Π** . Če ta integral izračunamo, dobimo funkcijo, prikazano na sliki 33, imenovano *normalizirana funkcija sinc*:

$$\hat{\Pi}(\mu) = \frac{\sin(\pi x)}{\pi x} = \text{sinc}(x)$$

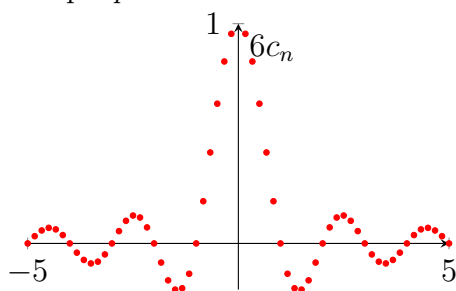
¹⁰<http://www.wolframalpha.com/>



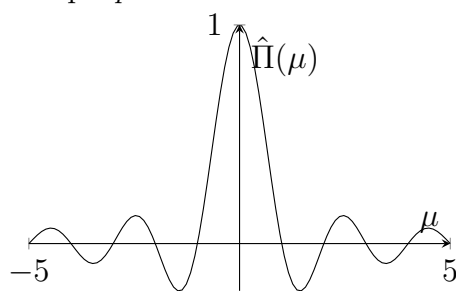
Slika 30: Skalirani Fourierjevi koeficienti pri $p = 2$



Slika 31: Skalirani Fourierjevi koeficienti pri $p = 4$



Slika 32: Skalirani Fourierjevi koeficienti pri $p = 6$



Slika 33: Fourierjeva transformacija $\hat{\Pi}$ pravokotnega impulza

Tako kot za funkcijo Π lahko tudi za katero koli drugo neperiodično funkcijo $f(x)$ poiščemo njeno Fourierjevo transformacijo:

$$\hat{f}(\mu) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\mu x} dx \quad (2.4)$$

Naj poudarim, da je – čeprav smo v primeru funkcije Π dobili realno funkcijo – v splošnem Fourierjeva transformacija **kompleksna funkcija**.

Pomen in razlaga Fourierjeve transformacije Tako kot smo periodično funkcijo zapisali s Fourierjevo vrsto, katere koeficienti posameznih členov so nam dajali informacijo o prisotnosti različnih frekvenc, iz katerih je sestavljena funkcija, tako nam tudi Fourierjeva transformacija prikazuje **delež posameznih frekvenc**, ki sestavljajo funkcijo. Vrednost transformacije $\hat{f}(\mu)$ pri danem μ nam daje informacijo o prisotnosti frekvence μ v prvotni funkciji.

Za Fourierjevo transformacijo rečemo, da se nahaja v **frekvenčni domeni**, medtem ko je prvotna funkcija v **prostorski domeni**.

Poleg mnogih občudovanja vrednih lastnosti Fourierjeve transformacije, za katere ni dovolj prostora, pa je (vsaj za nas) najpomembnejša **inverzna Fourierjeva**

transformacija, ki nam omogoča, da funkcijo, predstavljeno v frekvenčni domeni, ponovno prestavimo nazaj v njeno prostorsko domeno.

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\mu) e^{i2\pi\mu x} d\mu \quad (2.5)$$

V tem trenutku imamo občutek, da digitalna obdelava slik pravzaprav ni več tema besedila, a resnica je prav nasprotna! Opremljeni s Fourierjevo transformacijo se sedaj lotimo uporabe tega orodja pri obdelavi fotografij. Žal takoj naletimo na dve težavi. Prvi: funkcija slike je diskretna in ne zvezna; ter drugi: digitalna slika je funkcija dveh ravninskih spremenljivk. Brez izpeljevanja bom zato zapisal **diskretno Fourierjevo transformacijo (DFT)** ene in dveh spremenljivk ter njuni inverzni transformaciji.

$$\hat{f}(u) = \sum_{x=0}^{M-1} f(x) e^{-i2\pi \frac{ux}{M}}$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} \hat{f}(u) e^{i2\pi \frac{ux}{M}}$$

$$\hat{\mathcal{S}}(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mathcal{S}(x, y) e^{-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad (2.6)$$

pri čemer je $\mathcal{S}(x, y)$ funkcija slike, u in v diskretni spremenljivki v frekvenčni domeni, x in y diskretni spremenljivki v prostorski domeni ter M in N dimenziji matrike slike (M število vrstic in N število stolpcev).

Inverzna diskretna Fourierjeva transformacija dveh spremenljivk (IDFT):

$$\mathcal{S}(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{\mathcal{S}}(u, v) e^{i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad (2.7)$$

Vidimo lahko, da se Fourierjeva transformacija in njena inverzna funkcija zelo malo razlikujeta – različen je predznak eksponenta eksponentne funkcije, inverzna transformacija pa vsebuje tudi koeficient $\frac{1}{MN}$.

2.8.3 Interpretacija Fourierjeve transformacije in filtriranje

Fourierjeva vrsta nam prikazuje frekvenčni spekter slike – količine posameznih frekvenc, ki sestavljajo sliko. Ker bomo sliko Fourierjeve transformacije kasneje centriral (glej razdelek 3.2.1), bodo na robovih slike prikazane visoke frekvence, ki

nakazujejo ostre robove in podrobnosti, medtem ko bodo blizu središča nizke frekvence, ki zaznamujejo počasna spreminjanja intenzitete. Filtriranje v frekvenčni domeni nam tako omogoči, da odstranimo ali obdržimo samo določene frekvence in poteka tako, da izvorno sliko v prostorski domeni transformiramo v frekvenčno domeno, jo pomnožimo s filtrom $H(u, v)$ in pretvorimo nazaj v prostorsko domeno. Vrednosti slike filtra so tokrat izjemoma realna števila na intervalu $[0, 1]$, tako da množenje s Fourierjevo transformacijo nikoli ne vrne absolutno večje vrednosti.

2.8.4 Nizkoprepustni filter

V angleščini imenovan *low-pass filter* nam omogoča, da odstranimo visoke frekvence in s tem tudi ostre robove ter podrobnosti na sliki. Končni rezultat je zmeščana oziroma zamegljena slika.

Idealni nizkoprepustni filter Filter generiramo tako, da izberemo mejno frekvenco d_0 in na „črni“ podlagi izrišemo „bel“ krog s polmerom d_0 – črna barva je v filtru predstavljena z vrednostjo 0, bela z vrednostjo 1. Na končni sliki, mehčani z idealnim nizkopasovnim filtrom, se pojavi vzorec, značilen za idealne filtre, imenovan *ringing effect* (nihajoči odziv).

$$H(u, v) = \begin{cases} 0 & ; d(u, v) > d_0 \\ 1 & ; d(u, v) \leq d_0 \end{cases}$$

Gaussov nizkoprepustni filter Zaradi neželenega učinka, ki ga povzroči idealni filter, obstajajo tudi drugi filtri. Eden takih je Gaussov, pri katerem ne obstaja mejna frekvenca, temveč je prehod mehkejši. Filter generiramo tako, da si izberemo varianco σ , nato pa vrednost posameznega piksla izračunamo s funkcijo, podobno gostoti verjetnosti za normalno (Gaussovo) porazdelitev. Naša „statistična spremenljivka“ je v tem primeru razdalja piksla od središča, torej $d(u, v) = \sqrt{(u - M)^2 + (v - N)^2}$:

$$H(u, v) = e^{-\frac{d^2(u, v)}{2\sigma^2}}$$

Če za mehčanje uporabimo ta filter, ne pride do prej omenjenega učinka nihajočega odziva, zato je tudi primernejši za praktično uporabo.

2.8.5 Visokoprepustni filter

Z visokoprepustnim filtrom (ang. *high-pass filter*) izločimo iz slike vse počasne prehode in ohranimo samo ostre robove in detajle. Tudi tukaj lahko uporabimo

idealni ali Gaussov filter¹¹.

Idealni visokoprepustni filter

$$H(u, v) = \begin{cases} 1 & ; d(u, v) > d_0 \\ 0 & ; d(u, v) \leq d_0 \end{cases}$$

Gaussov visokoprepustni filter

$$H(u, v) = 1 - e^{-\frac{d^2(u, v)}{2\sigma^2}}$$

2.8.6 Pasovnozaporni in pasovnoprepustni filtri

S pasovnozapornimi filtri odstranimo samo frekvence v določenem območju, medtem ko s pasovnoprepustnimi filtri odstranimo vse ostale frekvence.

Pasovnozaporni filter

Idealni: določimo mejni frekvenci d_0 in d_1 tako, da $d_0 < d_1$

$$H(u, v) = \begin{cases} 1 & ; d(u, v) > d_1 \vee d(u, v) < d_0 \\ 0 & ; d_0 \leq d(u, v) \leq d_1 \end{cases}$$

Gaussov: določimo premik od središča d_0 in varianco σ

$$H(u, v) = e^{-\frac{(d(u, v) - d_0)^2}{2\sigma^2}}$$

Pasovnoprepustni filter

Idealni: določimo mejni frekvenci d_0 in d_1 tako, da $d_0 < d_1$

$$H(u, v) = \begin{cases} 0 & ; d(u, v) > d_1 \vee d(u, v) < d_0 \\ 1 & ; d_0 \leq d(u, v) \leq d_1 \end{cases}$$

Gaussov: določimo premik d_0 in varianco σ

$$H(u, v) = 1 - e^{-\frac{(d(u, v) - d_0)^2}{2\sigma^2}}$$

2.8.7 Poljubno filtriranje

Filtriramo lahko tudi tako, da izberemo samo poljubna območja v frekvenčni domeni, ki jih želimo obdržati ali ohraniti.

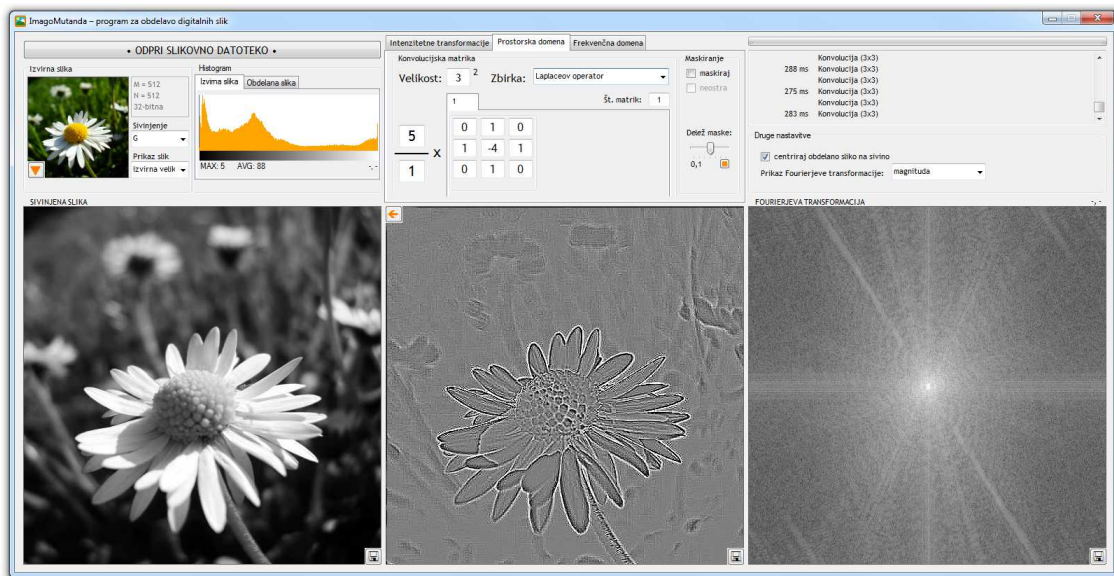
¹¹Sicer obstaja tudi Butterworthov filter, ki je vmesna stopnja med idealnim in Gaussovim filtrom.

3 RAČUNALNIŠKA IMPLEMENTACIJA

Program sem izdelal v razvojnem okolju **Visual Basic Studio 2012 Express** v programskem jeziku Visual Basic z ogrodjem *.NET*. Za računanje s kompleksnimi števili sem uporabil že vključeno knjižnico *Numerics.Complex* z izdelanimi funkcijami za osnovne računske operacije med kompleksnimi števili in izračun kompleksne eksponentne funkcije.

Računsko zahtevnejše oziroma dolgotrajnejše procedure, kot sta (inverzna) Fourierjeva transformacija in konvolucija, se izvedejo asinhrono, tj. v novi programski niti, z uporabo kontrolnika *BackgroundWorker*.

V opisih procedur uporabljam spremenljivko M kot število vrstic ali kot število elementov zaporedja (npr. število pikslov v eni vrstici).



Slika 34: Zaslonski posnetek programa

3.1 Računanje Fourierjeve transformacije

Najprej bomo Fourierjevo transformacijo dveh spremenljivk nekoliko preoblikovali. Če uporabljamo osnovno obliko, potem se moramo za vsak u in v v frekvenčni domeni sprehoditi skozi vse x in y v prostorski domeni – to lahko spremenimo. Najprej eksponentno funkcijo nekoliko razpišemo:

$$\begin{aligned}\hat{\mathcal{S}}(u, v) &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mathcal{S}(x, y) e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} = \\ &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mathcal{S}(x, y) e^{-\frac{i2\pi ux}{M}} e^{-\frac{i2\pi vy}{N}}\end{aligned}$$

Ker je $e^{-\frac{i2\pi ux}{M}}$ pri istih x konstanten, ga lahko izpostavimo:

$$\hat{\mathcal{S}}(u, v) = \sum_{x=0}^{M-1} \left(e^{-\frac{i2\pi ux}{M}} \sum_{y=0}^{N-1} \mathcal{S}(x, y) e^{-\frac{i2\pi vy}{N}} \right)$$

Notranjo vrsto sedaj zapišemo kot funkcijo spremenljivk x in v .

$$\hat{\mathcal{S}}_1(x, v) = \sum_{y=0}^{N-1} \mathcal{S}(x, y) e^{-\frac{i2\pi vy}{N}}$$

Vidimo, da je to pravzaprav enostavna Fourierjeva transformacija funkcije ene spremenljivke, in sicer je to transformacija vrstice slike z indeksom x . Če sedaj zapišemo še vse skupaj:

$$\hat{\mathcal{S}}(u, v) = \sum_{x=0}^{M-1} e^{-\frac{i2\pi ux}{M}} \hat{\mathcal{S}}_1(x, v) \quad (3.1)$$

Vidimo, da smo ponovno dobili Fourierjevo transformacijo, in sicer za vsako vrstico. Tako lahko Fourierjevo transformacijo naše celotne slike izračunamo tako, da najprej izračunamo Fourierjevo transformacijo za vsako vrstico posebej, potem pa iz dobljenih rezultatov izračunamo še Fourierjevo transformacijo za stolpce¹².

¹²Transformaciji s tako lastnostjo pravimo **separabilna**.

3.1.1 Hitra Fourierjeva transformacija (FFT)

Če pogledamo osnovno formulo za diskretno Fourierjevo transformacijo v dveh dimenzijah:

$$\hat{f}(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

vidimo, da zahteva izjemno veliko računske moči, saj moramo za vsak piksel izračunati MN členov. Za celotno sliko to znese kar M^2N^2 členov, kar je izjemno veliko – za srednje veliko sliko velikosti 600×600 pikslov moramo izračunati kar $129\,600\,000\,000 \doteq 10^{11}$ členov! To je ogromno dela tudi za današnje računalnike. K sreči lahko algoritem računanja diskretne Fourierjeve transformacije izboljšamo na različne načine, kar je prvi naredil že Gauss, ko je računal tirnico takrat opazovanega asteroida Palasa. Seveda njegove domiselnosti dolgo niso uvideli in šele nekaj let po objavi članka avtorjev Cooley in Tukey leta 1965 so sprevideli, da je njuno „novoodkrita“ metodo že pred 160 leti odkril Gauss (OSGOOD, 2007).

Tukaj bom opisal in v programu tudi uporabil metodo zaporednega razpolavljanja (ang. *successive-doubling method*), imenovano tudi *radix-2 FFT*, ki je posebna oblika algoritma Cooley-Tukey z zahtevo, da je izvirna slika v velikosti $M \times N$, pri čemer sta širina in višina potenci števila 2 ($M = 2^m$ in $N = 2^n$).

Ker enačba 3.1 prikazuje, da lahko izračunamo Fourierjevo transformacijo slike z računanjem stolpcev in vrstic posebej, se moramo pravzaprav osredotočiti samo na optimizacijo računanja Fourierjeve transformacije funkcije ene spremenljivke. Osvežimo:

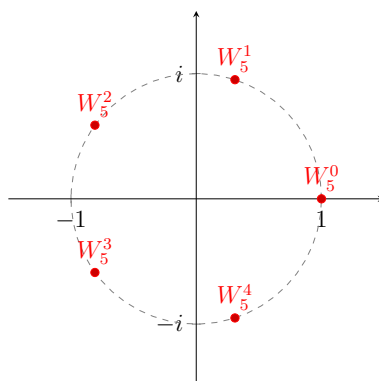
$$\hat{f}(u) = \sum_{x=0}^{M-1} f(x) e^{-\frac{i2\pi ux}{M}}$$

pri čemer $u = 0, 1, 2, \dots, M - 1$.

Postavili smo začetni pogoj $M = 2^m$. M je torej zagotovo sodo število in ga lahko zapišemo kot $M = 2K$. Sedaj zaradi jasnosti in preglednosti uvedemo nov simbol, ki predstavlja kompleksno eksponentno funkcijo, in sicer:

$$W_M = e^{-\frac{i2\pi}{M}}$$

Dovolil si bom majhen ekskurz o zanimivosti, ki se nam že ves čas pojavlja pred očmi in je z uvedbo novega simbola še jasneje prikazana. W_M pravzaprav predstavlja enega izmed **M -tih korenov enote**. Koreni enote so rešitve krožno-delitvene (binomske) enačbe $z^M = 1$ in v kompleksni ravnini ležijo na enotski krožnici. Kasneje bom pokazal tudi nekaj zanimivih lastnosti teh števil.



Slika 35: Vseh 5 kompleksnih rešitev enačbe $z^5 = 1$.

Sedaj ponovno zapišemo Fourierjevo transformacijo s tem simbolom:

$$\hat{f}(u) = \sum_{x=0}^{2K-1} f(x)W_M^{ux}$$

Zaradi jasnosti zapišimo konkretni primer: naj bo $K = 4 \Rightarrow M = 8$

$$\hat{f}(u) = f(0)W_8^0 + f(1)W_8^u + f(2)W_8^{2u} + f(3)W_8^{3u} + f(4)W_8^{4u} + f(5)W_8^{5u} + f(6)W_8^{6u} + f(7)W_8^{7u}$$

Trik, ki ga sedaj uporabimo, je združitev členov na sodih mestih in členov na lihih mestih:

$$\begin{aligned} \hat{f}(u) &= (f(0)W_8^0 + f(2)W_8^{2u} + f(4)W_8^{4u} + f(6)W_8^{6u}) + \\ &\quad + (f(1)W_8^u + f(3)W_8^{3u} + f(5)W_8^{5u} + f(7)W_8^{7u}) = \\ &= \underbrace{\sum_{x=0}^3 f(2x)W_8^{u(2x)}}_{sodi} + \underbrace{\sum_{x=0}^3 f(2x+1)W_8^{u(2x+1)}}_{lihi} \end{aligned}$$

Če to zapišemo v splošni obliki za poljubno funkcijo, dobimo:

$$\hat{f}(u) = \sum_{x=0}^{K-1} f(2x)W_{2K}^{u(2x)} + \sum_{x=0}^{K-1} f(2x+1)W_{2K}^{u(2x+1)}$$

Posvetimo se koeficientu $W_{2K}^{u(2x)}$:

$$W_{2K}^{u(2x)} = e^{\frac{-i2\pi \cdot 2ux}{2K}} = e^{\frac{-i2\pi \cdot ux}{K}} = W_K^{ux}$$

Torej:

$$\hat{f}(u) = \sum_{x=0}^{K-1} f(2x)W_K^{ux} + \sum_{x=0}^{K-1} f(2x+1)W_K^{ux}W_{2K}^u$$

Sedaj posamezni vrsti zapišemo kot novi funkciji, in sicer $F_S(u) = \sum_{x=0}^{K-1} f(2x)W_K^{ux}$ za sode člene in $F_L(u) = \sum_{x=0}^{K-1} f(2x+1)W_K^{ux}$ za lihe člene:

$$\hat{f}(u) = F_S(u) + F_L(u)W_{2K}^u$$

Skoraj smo že pri koncu. Najprej izpeljimo še dve zanimivi lastnosti simbola W_K :

$$\boxed{W_M^{u+M} = W_M^u} \quad (3.2)$$

Dokaz 1

$$W_M^{u+M} = e^{\frac{-i2\pi \cdot (u+M)}{M}} = e^{\frac{-i2\pi u}{M}} \cdot e^{-i2\pi}$$

Ker:

$$e^{-i2\pi u} = \cos(-2\pi u) + i \sin(-2\pi u) = \cos(2\pi u) - i \sin(2\pi u) = 1 - 0i = 1 \text{ za vsak } u \in \mathbb{Z}$$

sledi:

$$W_M^{u+M} = W_M^u$$

V koordinatnem sistemu bi to pomenilo vrtenje kompleksnega števila za polni kot okoli središča – rezultat je seveda prvotno število.

In še druga lastnost:

$$\boxed{W_{2M}^{u+M} = -W_{2M}^u} \quad (3.3)$$

Dokaz 2

$$W_{2M}^{u+M} = e^{\frac{-i2\pi \cdot (u+M)}{2M}} = e^{\frac{-i2\pi u}{2M}} \cdot e^{\frac{-i2\pi M}{2M}} = e^{\frac{-i\pi u}{M}} \cdot e^{-i\pi}$$

Ker:

$$e^{-i\pi} = \cos(\pi) + i \sin(\pi) = -1 + 0i = -1$$

sledi:

$$W_{2M}^{u+M} = -e^{\frac{-i\pi u}{M}} = -e^{\frac{-i2\pi u}{2M}} = -W_{2M}^u$$

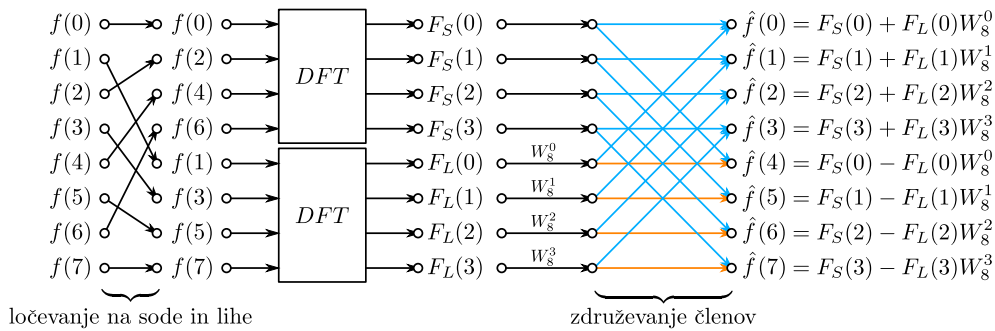
Grafično to pomeni vrtenje kompleksnega števila okoli središča za pol kroga; rezultat je nasprotno število.

Z danima lastnostma se podajmo v iskanje „premika“ $\hat{f}(u+K)$.

$$\begin{aligned}
\hat{f}(u+K) &= F_S(u+K) + F_L(u+K)W_{2K} = \\
&= \sum_{x=0}^{K-1} f(2x)W_K^{(u+K)x} + \sum_{x=0}^{K-1} f(2x+1)W_K^{(u+K)x}W_{2K}^{u+K} = \\
&= \sum_{x=0}^{K-1} f(2x)W_K^{ux} - \sum_{x=0}^{K-1} f(2x+1)W_K^{(u+K)x}W_{2K}^u = \\
&= F_S(u) - F_L(u)W_{2K}^u
\end{aligned}$$

$W_K^{(u+K)x}$ iz druge vrstice smo zamenjali z $W_K^{(u+K)x} = (W_K^{u+K})^x = W_K^{ux}$, W_{2K}^{u+K} pa z $W_{2K}^{u+K} = -W_{2K}^u$.

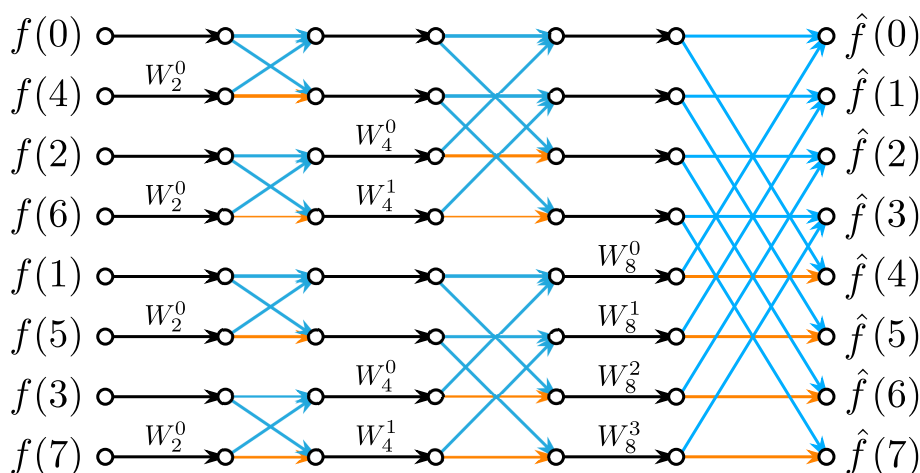
Čeprav na prvi pogled ni jasno, kaj smo s tem dosegli, lahko s podrobnejšim razmislekom pridemo do zanimivega uvida. Za diskretno Fourierjevo transformacijo M elementov elemente najprej razdelimo na dve polovici: tiste s sodimi in tiste z lihimi koeficienti. Sedaj izračunamo Fourierjevo transformacijo posebej za sode člene ($F_S(u)$) in lihe člene ($F_L(u)$). Posamezni končni $\hat{f}(u)$ sedaj dobimo s seštevanjem $F_S(u) + F_L(u)W_M^u$. Pri prvi polovici $\hat{f}(u)$ težav nimamo, saj u teče od 0 do K , pri drugi polovici pa upoštevamo zgoraj izpeljano periodičnost $\hat{f}(u+K) = F_S(u) - F_L(u)W_M^u$. Celoten postopek se najlažje predstavi s pomočjo diagrama. V diagramu je prikazan potek hitre Fourierjeve transformacije za 8 diskretnih elementov. Puščice prikazujejo potek posameznih vrednosti. Modra puščica označuje seštevanje, oranžna pa odštevanje. Število, zapisano nad puščico, predstavlja faktor množenja.



Slika 36: DFT osmih elementov z izboljšavo razdelitve na sode in lihe člene

Vsekakor se tukaj zadeva še ne konča. Metodo delitve elementov na sode in lihe¹³ lahko ponovno uporabimo, in sicer pri računanju DFT štirih elementov, ki jih ponovno razdelimo na sode člene in lihe člene. Tako dobimo skupaj štiri pare elementov, za katere moramo izračunati diskretno Fourierjevo transformacijo.

¹³Sode in lihe glede na njihov položaj, seveda.



Slika 37: Popolna hitra Fourierjeva transformacija (FFT) osmih elementov

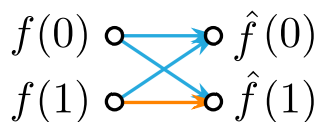
Algoritem je skrčen do te mere, da računamo diskretno Fourierjevo transformacijo samo za dva elementa, potem pa dobljene vrednosti med seboj paroma kombiniramo. Prav to je namen hitre transformacije – namesto računanja transformacij za mnogo elementov jih razbijemo na posamezne pare. Tukaj se skriva tudi razlog za prvotni pogoj, da mora biti število elementov potenca števila 2 – le tako lahko vseh M elementov postopoma razbijemo na $\log_2 M$ parov.

S tem, ko smo prvotne elemente razdelili v pare, nam ostane samo računanje DFT za dva elementa, kar pa je zelo enostavna naloga. V tem primeru je $M = 2$:

$$\hat{f}(0) = \sum_{x=0}^1 f(x)W_2^0 = f(0)e^0 + f(1)e^0 = f(0) + f(1)$$

$$\hat{f}(1) = \sum_{x=0}^1 f(x)W_2^1 = f(0)e^0 + f(1)e^{-i\pi} = f(0) - f(1)$$

Diskretno Fourierjevo transformacijo dveh elementov torej izvedemo tako, da oba elementa seštejemo – dobimo $\hat{f}(0)$ –, nato pa drugega odštejemo od prvega – dobimo $\hat{f}(1)$.



Slika 38: DFT dveh elementov

Zadnji korak naše hitre Fourierjeve transformacije je zgolj še preizkus: jo upravičeno imenujemo *hitra* Fourierjeva transformacija? Preverimo.

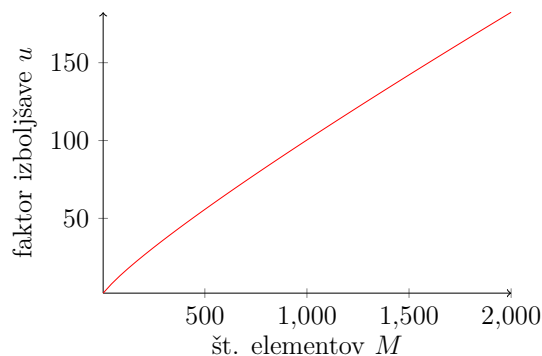
Pri računanju DFT po osnovni formuli 2.6, smo vsak končni element dobili z vsoto M členov, vsak od teh členov pa je produkt nekega začetnega elementa in nekega W_M . Za vsak element smo torej izvedli M množenj in M seštevanj – za celotno DFT to znese M^2 množenj in M^2 seštevanj oz. $2M^2$ operacij.

Pri FFT razdelimo celoten postopek na $\log_2 M$ posameznih faz, vsaka faza pa je sestavljena iz M seštevanj in $\frac{M}{2}$ množenj, vse skupaj torej $M \log_2 M$ seštevanj in $M \log_2 M$ množenj oz. $2M \log_2 M$ operacij.

Učinkovitost metode FFT (u) bomo dobili tako, da bomo število operacij pri (navadni) DFT delili s številom operacij pri FFT, torej

$$u(M) = \frac{2M^2}{2M \log_2 M} = \frac{M}{\log_2 M}$$

Za DFT 32 elementov bi pri navadni metodi potrebovali $2 \cdot 32^2 = 2048$ računskih operacij, medtem ko bi z metodo FFT potrebovali samo $2 \cdot 32 \cdot \log_2 32 = 320$ operacij, kar je približno 6-krat manj! Z naraščanjem števila elementov narašča tudi učinkovitost FFT, tako da je učinkovitost pri računanju DFT 512 elementov (DFT = 524288, FFT = 9216) že približno 57-krat izboljšana.



Slika 39: Graf faktorja izboljšave u hitre Fourierjeve transformacije v odvisnosti od števila elementov M

3.2 Osnovne procedure

Računanje primitivnih korenov enote W_M Množico koeficientov $W_M^n = e^{-\frac{i2\pi n}{M}}$ za dani M , pri čemer teče n od 0 do M , program izračuna tako, da najprej izračuna konstantni faktor $e_0 = e^{-\frac{i2\pi}{M}}$, nato pa za vsak n izračuna $W_M^n = e_0 \cdot e^n$. Množico koeficientov za inverzno Fourierjevo transformacijo dobimo tako, da pri prej dobljeni množici ohranimo prvi koeficient ($W_M^0 = 1$), ostale pa postavimo v obratni vrstni red. Če si te koeficiente predstavljamo na enotski krožnici, si sledijo prvotni od števila 1 v smeri urinega kazalca, koeficienti za inverzno transformacijo pa potekajo od 1 v nasprotni smeri urinega kazalca.

Ničelno zapolnjevanje Ker mora število elementov M za uspešno izvedbo FFT biti potenca števila 2, moramo krajše nize popraviti tako, da jim na koncu dodamo prazne elemente – števila 0. Število dodanih ničel N_0 je enako razliki med najbližjo višjo potenco 2 in številom M :

$$N_0 = 2^{\lceil \log_2 M \rceil} - M$$

pri čemer je $\lceil \log_2 M \rceil$ najbližje naravno število, za katerega je $\lceil \log_2 M \rceil \geq \log_2 M$.

Bitreverzno zaporedje V prvem koraku FFT izvajamo transformacijo na dveh elementih – posamezne skupinice dveh elementov dobimo tako, da vse elemente zaporedoma delimo na sode in lihe glede na njihov položaj v zaporedju. Izkaže se, da so po takem razvrščanju elementi razvrščeni na prav poseben, zanimiv način – njihovi obrnjeni binarni zapisi so razvrščeni v naraščajoče zaporedje. V tabeli je prikazan začetni razpored 8 elementov za FFT. Indekse elementov ($n_{(10)}$) od 0 do 7 najprej zapišemo v binarni obliki ($n_{(2)}$), nato zaporedje ničel in enic obrnemo ($n'_{(2)}$) in pretvorimo nazaj v desetiški sestav ($n'_{(10)}$). FFT bomo sedaj začeli tako, da bomo izračunali Fourierjeve transformacije za pare (0, 4), (2, 6), (1, 5), (3, 7) ...

$n_{(10)}$	$n_{(2)}$	$n'_{(2)}$	$n'_{(10)}$
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Tabela 3: Bitreverzno zaporedje 8 elementov

3.2.1 Prikazovanje Fourierjeve transformacije

Rezultat dvodimenzionalne Fourierjeve transformacije slike velikosti $M \times N$ je matrika iste velikosti. V našem primeru prvotna slika predstavlja posamezne intenzitete pikslov – vse vrednosti so torej realna števila –, Fourierjeva transformacija pa vrne matriko kompleksnih števil. Ker je zelo prikladno tudi to matriko prikazati kot digitalno sliko, moramo na tej matriki ponovno izvesti proces digitalizacije. Slika je že diskretizirana, preostane nam le še kvantizacija. Ker so vrednosti pikslov kompleksna števila, se moramo odločiti za eno izmed spodnjih realnih vrednosti, ki jih želimo prikazati na digitalni sliki Fourierjeve transformacije:

- realni del,
- kompleksni del,
- faza (argument) ali
- magnituda (absolutna vrednost).

Realnega in kompleksnega dela ne bom prikazoval, saj nista tako pomembna za nadaljnjo obdelavo.

Magnituda Najpogosteje prikazujemo **magnitudo** kompleksnega števila (imejemo jo tudi absolutna vrednost ali modul kompleksnega števila):

$$|z| = \sqrt{a^2 + b^2}$$

Ko izračunamo posamezne magnitude, ugotovimo, da je razpon vrednosti na sliki Fourierjeve transformacije zelo velik (največja in najmanjša magnituda se razlikujeta za več redov velikosti). Če bi te vrednosti kvantizirali linearno, kot smo storili pri kvantizaciji digitalne slike, nastale v fotoaparatu, bi izgubili veliko uporabnih podatkov (glej sliko 42), zato se raje odločimo za logaritemsko kvantizacijo, pri kateri je vrednost piksla enaka:

$$I_m = \xi(a \log_{10}(1 + |z|))$$

Člen 1 v logaritmandu je zagotovilo, da bo logaritem definiran – magnituda sicer ne more biti negativna, lahko pa je enaka 0. Koeficient a izberemo tako, da bo piksel z največjo magnitudo imel vrednost 255. S poznavanjem lastnosti Fourierjeve transformacije lahko predvidimo, da se bo piksel z največjo magnitudo nahajal na koordinatah $(0, 0)$ ¹⁴, zatorej:

$$a \log_{10}(1 + \mathcal{S}(0, 0)) = 255$$

¹⁴Po centriranju slike Fourierjeve transformacije se ta element nahaja na koordinatah $(\frac{M}{2}, \frac{N}{2})$.

$$a = \frac{255}{\log_{10}(1 + \mathcal{S}(0, 0))}$$

Funkcija ξ poskrbi za ustrezno zaokrožanje na najbližje celo število.

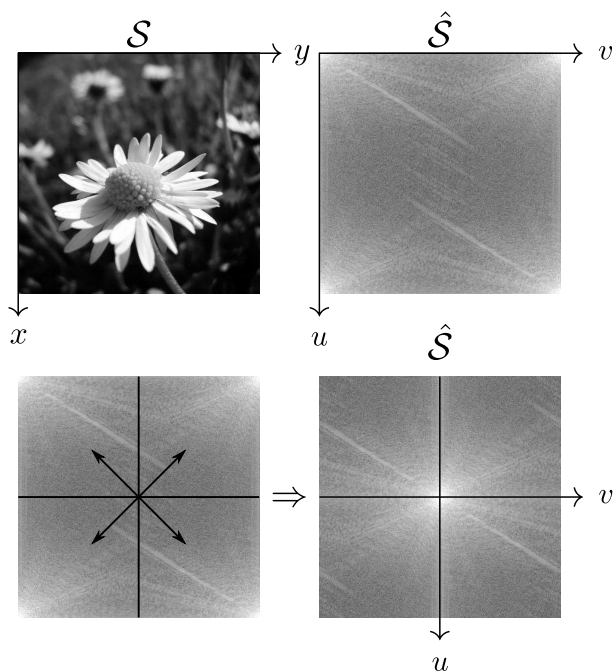
Faza Fazo (ali argument kompleksnega števila) program izračuna s pomočjo funkcije $\arctan2(y, x)$, ki vrne kot med pozitivnim krakom osi x in krajevnim vektorjem točke s koordinatama (x, y) . Funkcija je podobna funkciji *arkus tangens* (\arctan), le da upošteva predznaka koordinat, zato vrne ustrezen kot na intervalu $(-\pi, \pi]$. Vrednost piksla se izračuna kot:

$$I_\phi = \xi(40, 58 \arctan2(\Im(z), \Re(z)) + 127)$$

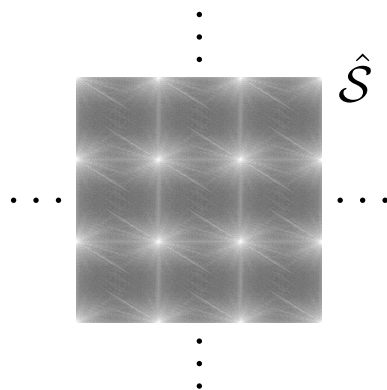
Faktor raztega 40, 58 je na dve decimalni mesti zaokrožena vrednost $\frac{255}{2\pi}$. Konstantni člen 127 je dodan – podobno kot pri obdelavi z Laplaceovim operatorjem –, da ne izgubimo negativnih vrednosti. Primer faze Fourierjeve transformacije je na sliki 42.

Centriranje Fourierjeve transformacije Ena izmed mnogih lastnosti diskretne Fourierjeve transformacije je njena **periodičnost**. Na sliki 40 je prikazana slika marjetice v prostorski domeni (levo zgoraj) in njena transformacija v frekvenčni domeni (desno zgoraj). Koordinatno izhodišče je pri obeh slikah v levem zgornjem kotu, vendar nam periodičnost Fourierjeve transformacije, prikazana na sliki 41, omogoča, da koordinatno izhodišče Fourierjeve transformacije premaknemo na sredino slike. To izvedemo tako, da sliko v frekvenčni domeni razdelimo na 4 enako velike pravokotnike (v tem primeru so kar kvadrati), stikajoče se v središču slike, in jih paroma diagonalno zamenjamo. Koordinatno izhodišče dobljene slike je sedaj v njenem središču, frekvenčni spremenljivki u in v pa se širita navzven.

Izračun histograma Program si vnaprej pripravi množico 256 elementov vrednosti 0 z indeksi od 0 do 255 (za vsako svetlostno intenziteto), nato se sprehodi skozi vsak piksel in prišteje 1 k elementu množice z indeksom njegove vrednosti. Po sprehodu skozi vse piksele dobimo absolutne frekvence svetlostnih intenzitet, ki jih statistično obdelamo (aritmetična sredina in modus). Slika histograma z višino h pikslov se izriše z zaporednim risanjem 256 navpičnih črt, katerih vodoravni položaj je enak svetlostni intenziteti i , višina pa je enaka $h \cdot n_i / n_{max}$, pri čemer je n_i število pikslov intenzitete i in n_{max} število pikslov modusa. Črta pri intenziteti z največjim številom pikslov (modus) bo tako enaka celotni višini histograma h .



Slika 40: Centriranje Fourierjeve transformacije



Slika 41: Periodičnost Fourierjeve transformacije

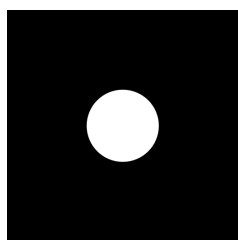
Binarne operacije med slikama Sliki se seštejeta/odštejeta/množita/delita tako, da se program sprehodi skozi vsak piksel in pri tem izvede ustrezno operacijo med istoležnima piksloma, nato dobljeno vrednost vstavi v funkcijo ξ (glej predpis 2.1).



Slika 42: Magnituda Fourierjeve transformacije v linearni kvantizaciji (levo) in logaritmski kvantizaciji (sredina) ter faza Fourierjeve transformacije (desno)

Generiranje idealnih filtrov Idealni filter se generira tako, da se na ustrezno podlago (belo/črno) izriše krog s središčem v sredini slike in polmerom r . Barva kroga je bela za nizkoprepustni filter in črna za visokoprepustni filter.

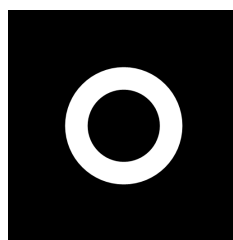
Pasovnoprepustni filter se generira tako, da se na nizkoprepustni filter s krogom, polmera r_2 , nariše nov črn krog s središčem v sredini slike in polmerom r , podobno se pasovnozaporni filter generira z risanjem belega kroga s polmerom r na obstoječi visokoprepustni filter s krogom r_2 .



Slika 43: Idealni nizkoprepustni filter velikosti 512×512 , $r = 80$



Slika 44: Idealni visokoprepustni filter velikosti 512×512 , $r = 80$



Slika 45: Idealni pasovnoprepustni filter velikosti 512×512 , $r = 80$, $r_2 = 130$



Slika 46: Idealni pasovnozaporni filter velikosti 512×512 , $r = 80$, $r_2 = 130$

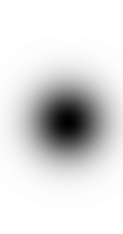
Generiranje Gaussovih filtrov Gaussov filter program generira tako, da z zmanjševanjem polmera r od polovice diagonale slike ($\frac{M\sqrt{2}}{2}$) do 0 s korakom 1 zaporedoma izrisuje kroge s središčem v sredini slike, katerih posamezne barve se izračunajo s formulo za Gaussov filter:

$$\text{barva}(r) = \begin{cases} e^{-\frac{(r-\mu)^2}{2\sigma^2}} & ; \text{ nizkoprepustni ali pasovnoprepustni} \\ 1 - e^{-\frac{(r-\mu)^2}{2\sigma^2}} & ; \text{ visokoprepustni ali pasovnozaporni} \end{cases}$$

Če izrisujemo nizkoprepustni ali visokoprepustni filter, je $\mu = 0$, pri pasovno-zapornem in pasovnoprepustnem filtru pa μ predstavlja premik Gaussove funkcije oziroma ciljno frekvenco, ki jo želimo ohraniti/odstraniti. Na filtru je μ viden kot razdalja od središča filtra do sredine kolobarja.



Slika 47: Gaussov nizkoprepustni filter velikosti 512×512 , $\sigma = 55$



Slika 48: Gaussov visokoprepustni filter velikosti 512×512 , $\sigma = 55$



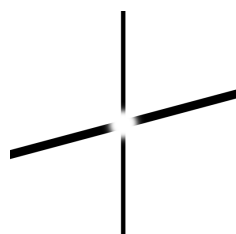
Slika 49: Gaussov pasovnoprepustni filter velikosti 512×512 , $\sigma = 25$, $\mu = 100$



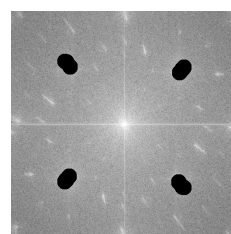
Slika 50: Gaussov pasovnozaporni filter velikosti 512×512 , $\sigma = 25$, $\mu = 100$

Generiranje poljubnih filtrov V program sem vključil dvojce različnih poljubnih filtrov, in sicer filtriranje s črtami in filtriranje z risanjem. Filtriranje s črtami poteka tako, da določimo število črt in njihove naklonske kote α (črte gredo skozi središče slike Fourierjeve transformacije; kot 0° predstavlja navpična črta), nato pa izberemo, ali naj bodo črte zaporne ali prepustne. Ob zapornih črtah je možno dodati tudi Gaussov nizkoprepustni filter z varianco σ .

Filtriranje z risanjem poteka tako, da z miško počrnimo sliko na območjih, ki jih želimo odstraniti. Pri tem lahko izberemo širino čopiča d , ki predstavlja premer krogov, ki se izrisujejo ob premikanju miške.



Slika 51: Filter z zapornimi črtami velikosti 512×512 , $\alpha_1 = 0^\circ$, $d_1 = 10$, $\alpha_2 = 75^\circ$, $d_2 = 20$, $\sigma = 15$



Slika 52: Slika Fourierjeve transformacije z dodanim filtrom, ustvarjenim z risanjem, velikosti 512×512

Konvolucija Izračun konvolucije slike velikosti $M \times N$ in konvolucijskega jedra velikosti $m \times n$ poteka tako, da program s sprehodom čez vse piksele slike in „premi-kanjem“ konvolucijskega jedra pri vsakem pikslu izračuna njegovo novo vrednost kot vsoto produktov istoležnih elementov:

$$I' = \sum_{s=-a}^a \sum_{t=-b}^b \mathcal{K}(s, t) \mathcal{S}(x - s, y - t)$$

Možno je izvesti tudi konvolucijo z več zaporednimi konvolucijskimi jedri, pri čemer se izvede konvolucija izvirne slike z vsakim posameznim jedrom, nato pa se slike seštejejo – to je uporabno pri Sobelovem in Prewittovem filtru za iskanje robov v obeh smereh, kjer uporabimo dve jedri.

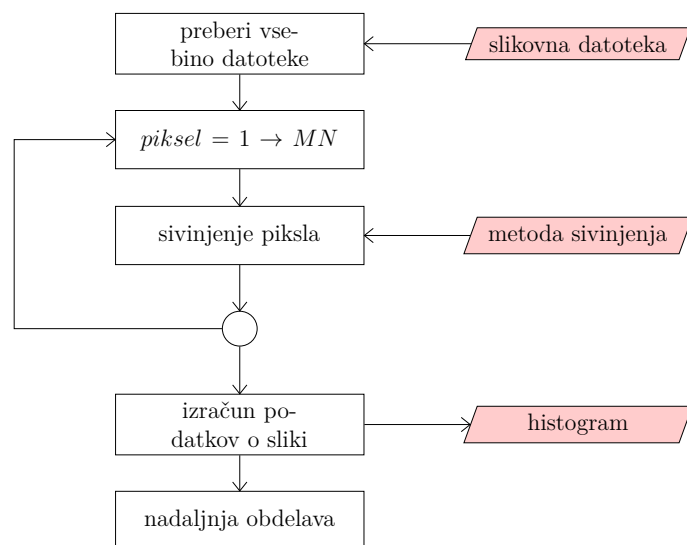
Program vsebuje zbirko že pripravljenih konvolucijskih jeder, in sicer:

- identiteta,
- premik 4 piksele v desno,
- svetleje $\left(\frac{4}{3}\right)$,
- temneje $\left(\frac{1}{2}\right)$,
- povprečje 3×3 ,
- povprečje 5×5 ,
- povprečje 7×7 ,
- Gaussovo povprečje 5×5 ,
- Laplaceov operator,
- Prewittovi operatorji x , y in xy ,
- Sobelovi operatorji x , y in xy .

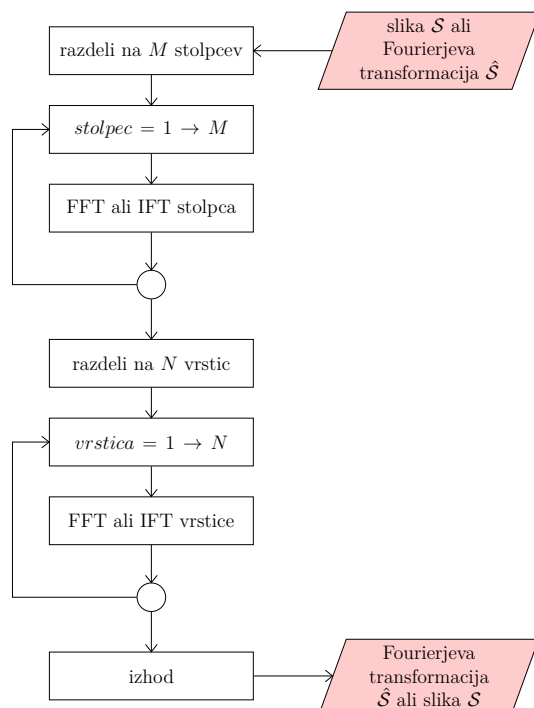
3.2.2 Diagrami potekov nekaterih algoritmov

S pomočjo poenostavljenih diagramov poteka sem prikazal delovanje določenih procedur programa. Pravokotniki prikazujejo posamezne korake procedure, paralelogrami prikazujejo vhodno/izhodne podatke, deltoidi pogojne stavke, krožci, povezani s pravokotnikom, pa zaznamujejo programsko zanko.

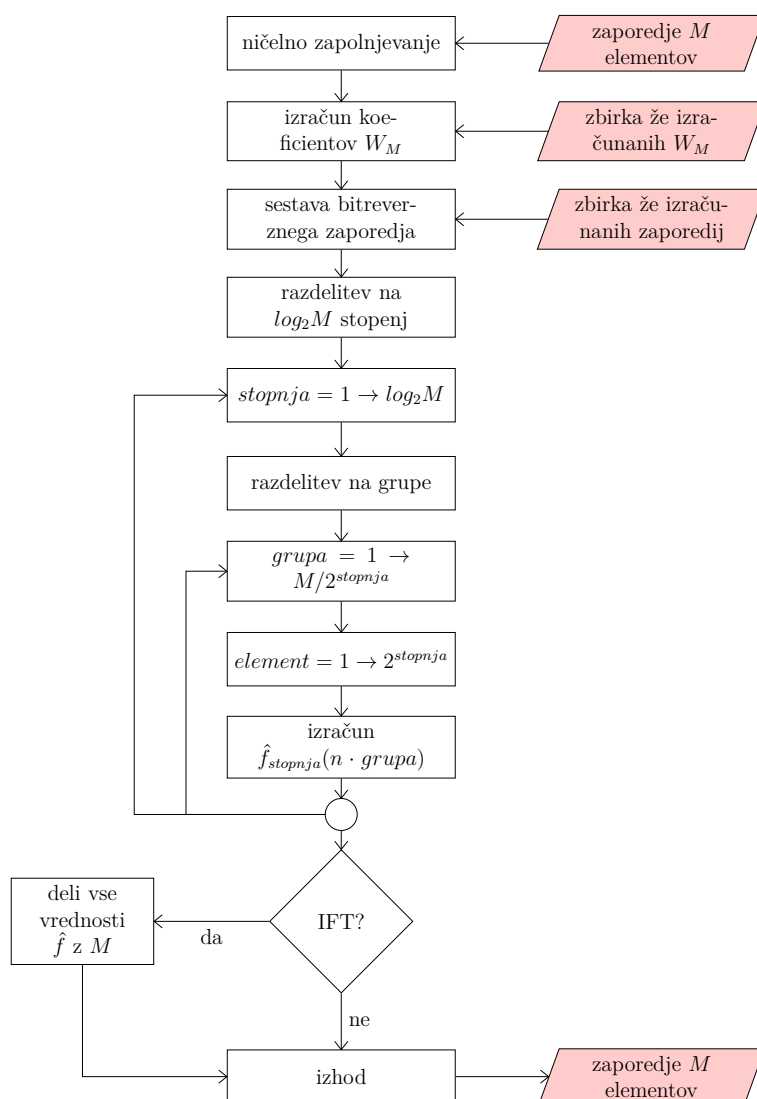
Za izračun Fourierjeve transformacije (FFT) in inverzne Fourierjeve transformacije (IFT) skrbi ena procedura, ki se razlikuje le v uporabljenih koeficientih W_M in zadnjem koraku, pri katerem se za inverzno Fourierjevo transformacijo vsi dobljeni elementi delijo s številom vseh elementov M .



Slika 53: Diagram poteka odpiranja slikovne datoteke



Slika 54: Diagram poteka (inverzne) hitre Fourierjeve transformacije slike



Slika 55: Diagram poteka (inverzne) hitre Fourierjeve transformacije M elementov

3.3 Primeri s programom obdelanih slik

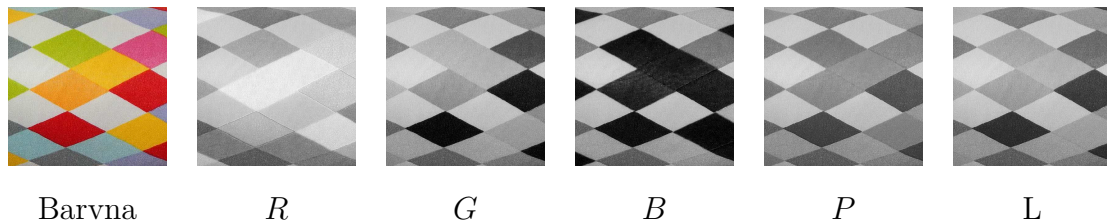
V nadaljevanju so prikazani rezultati digitalne obdelave slik z izdelanim programom.

3.3.1 Sivinjenje slike

Prva stopnja pri obdelavi slike iz slikovne datoteke je sivinjenje. V nadaljnjih napisih pod slikami je v oklepaju zapisana metoda sivinjenja izvirne barvne slike:

- P – aritmetična sredina treh barvnih kanalov,
- R – samo rdeči kanal,
- G – samo zeleni kanal,
- B – samo modri kanal,
- L – luminiscenca.

S sliko karirastega vzorca so prikazane različne metode sivinjenja.

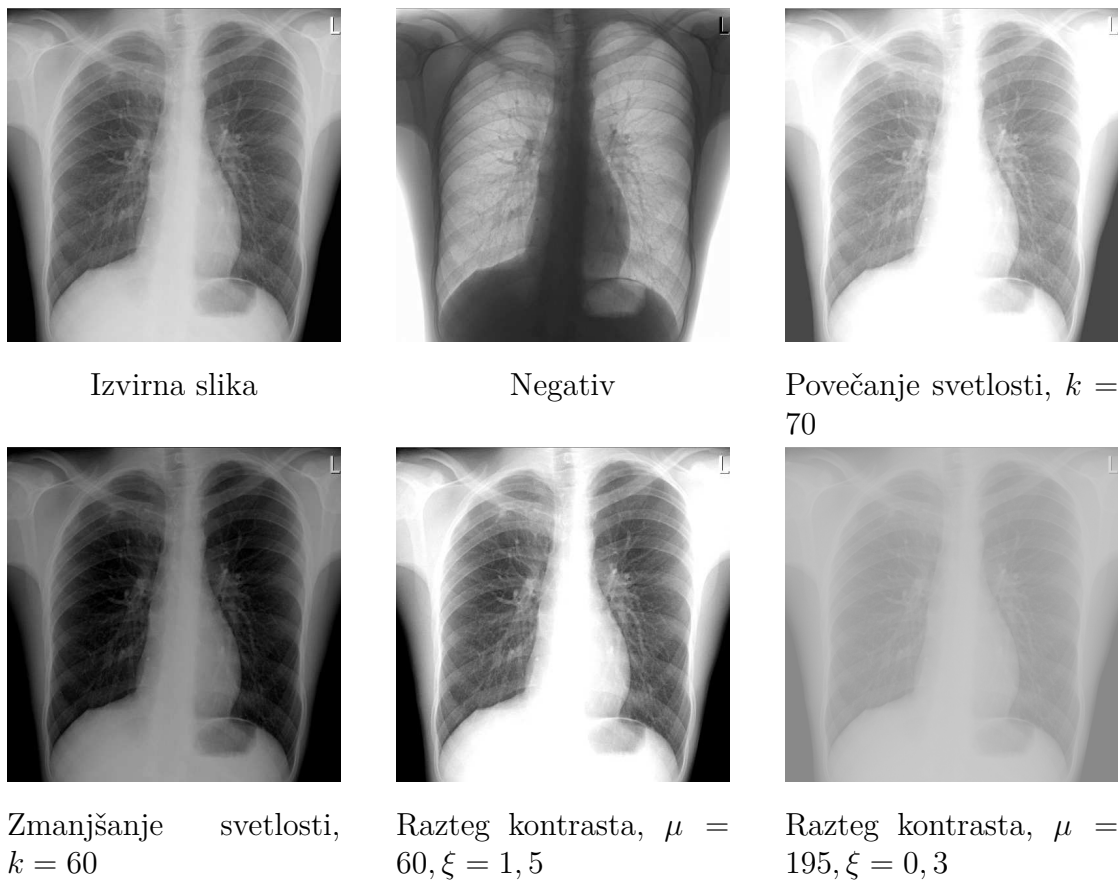


Slika 56: Različna izbira metode sivinjenja slike karirastega vzorca

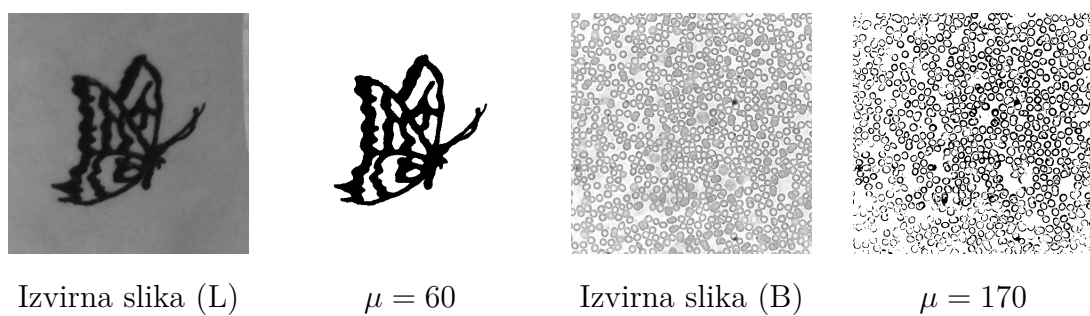
3.3.2 Intenzitetne transformacije

Na sliki rentgenograma prsnega koša so prikazane osnovne intenzitetne transformacije.

Upogovnanje slike s tatujem metulja nam na sliki ohrani samo tetoviranega metulja – táko črno-belo sliko bi lahko natisnili in uporabili kot predlogo pri izdelavi naslednjega tatuja ali pa jo vektorizirali. Upraženo sliko krvnega razmaza bi lahko poslali v nadaljnje postopke obdelave, kjer bi z morfološkimi in segmentacijskimi metodami izračunali število celic v vidnem polju, izmerili njihove velikosti in ugotovili morfološke nepravilnosti.



Slika 57: Intenzitetne transformacije rentgenograma prsnega koša



Slika 58: Upragovanje slike s tatujem metulja in slike krvnega razmaza

3.3.3 Obdelava v prostorski domeni

Mehčanje Digitalne slike v glavnem meglimo (mehčamo) zaradi umetniško-estetskih razlogov ali prikrivanja informacij na sliki, metoda pa je podlaga tudi za ostrenje slike s postopkom *unsharp masking*.



Izvirna slika (L)



Aritmetična sredina (5×5)



Gaussovo povprečenje (5×5)



Aritmetična sredina (7×7)

Slika 59: Povprečenje slike marjetice s konvolucijo

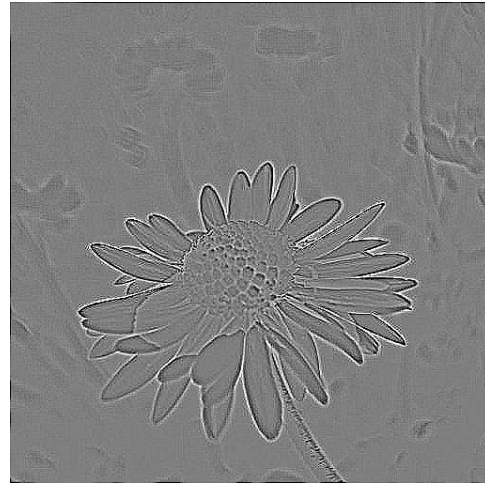
Iskanje robov in ostrenje Trije omenjeni operatorji – Laplaceov, Prewittov in Sobelov – se uporabljajo za zaznavanje robov objektov na sliki. Eno izmed področij uporabe so proizvodni procesi, katerih del so tudi avtomatski vizualni pregledi, ki preverjajo kakovost izdelkov in odkrivajo morebitne napake, nastale med proizvodnjo. Z zaznavanjem robov objektov in kasnejšo morfološko analizo lahko odkrijejo nepravilnosti.

Ostrenje slike je namenjeno izboljšanju prikaza ostrih robov in podrobnosti na

sliki, prav tako izboljša neostre slike, ki so nastale zaradi nepravilne nastavitve goriščne razdalje pri zajemanju slike.



Laplaceov operator, $c = 3$



Laplaceov operator, sredinjeno, $c = 3$

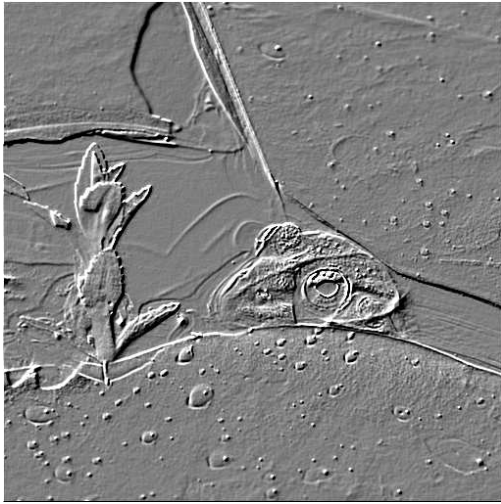


Laplaceov operator, maskirano, $c = 1$

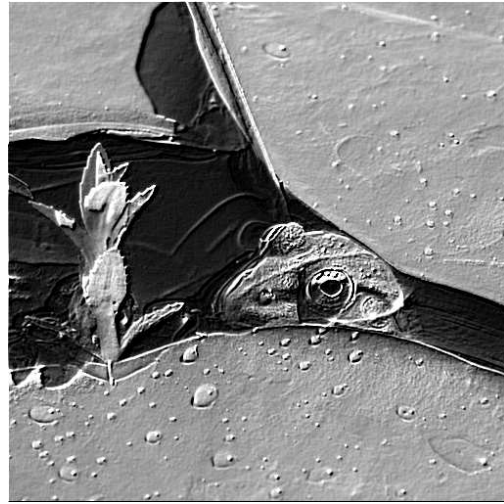


Laplaceov operator, maskirano, $c = 3$

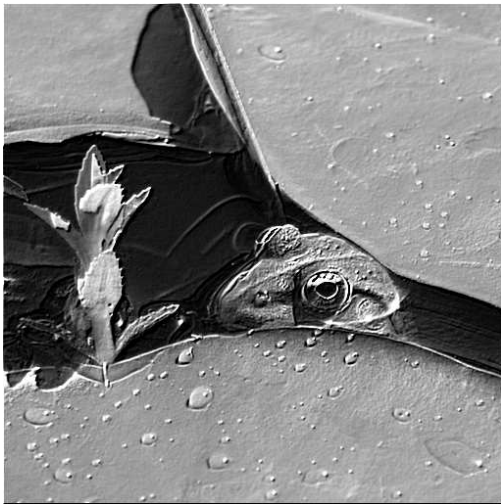
Slika 60: Laplaceov operator na sliki marjetice (L)



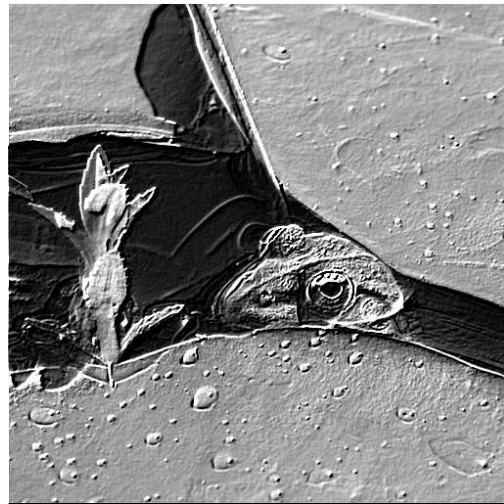
Prewittov operator xy , sredinjeno



Prewittov operator xy , maskirano, $c = 1$



Sobelov operator xy , maskirano, $c = 0,5$



Sobelov operator xy , maskirano, $c = 1$

Slika 61: Prewittov in Sobelov operator na sliki žabe (R)

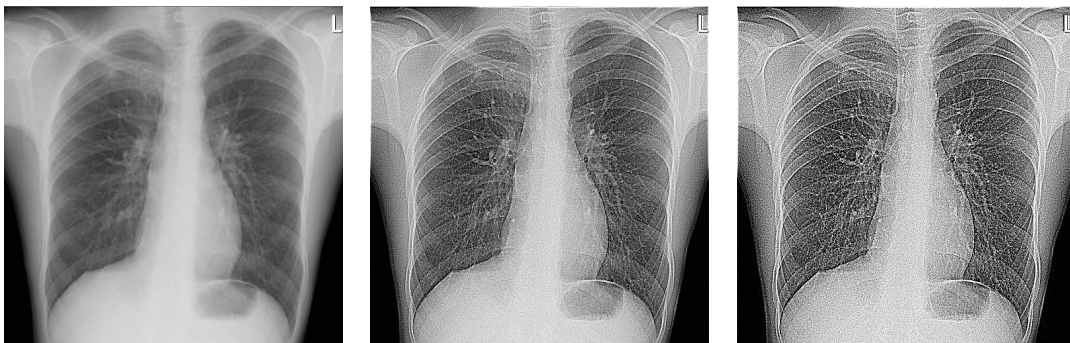


Izvirna slika (B)

UM, Gaussovo povprečje 5×5 , $c = 1$

UM, Gaussovo povprečje 5×5 , $c = 3$

Slika 62: Ostrenje z metodo *unsharp masking* (UM) na sliki Meseca



Izvirna slika

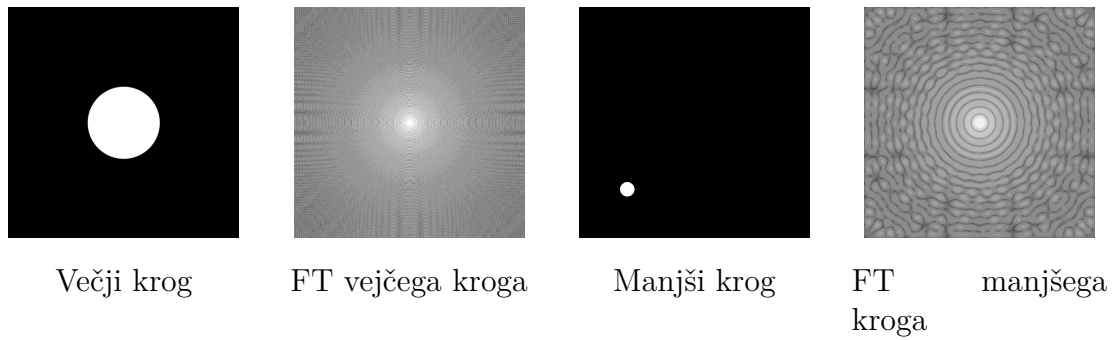
UM, Gaussovo povprečje 5×5 , $c = 1$

UM, Gaussovo povprečje 5×5 , $c = 2$

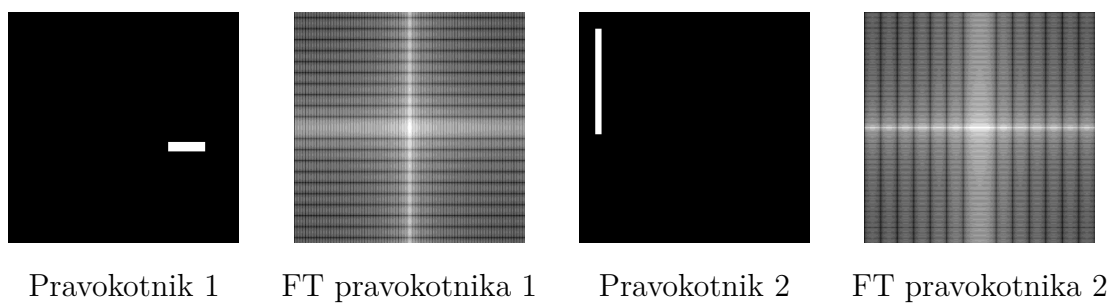
Slika 63: Ostrenje z metodo *unsharp masking* (UM) na rentgenogramu prsnega koša

3.3.4 Obdelava v frekvenčni domeni

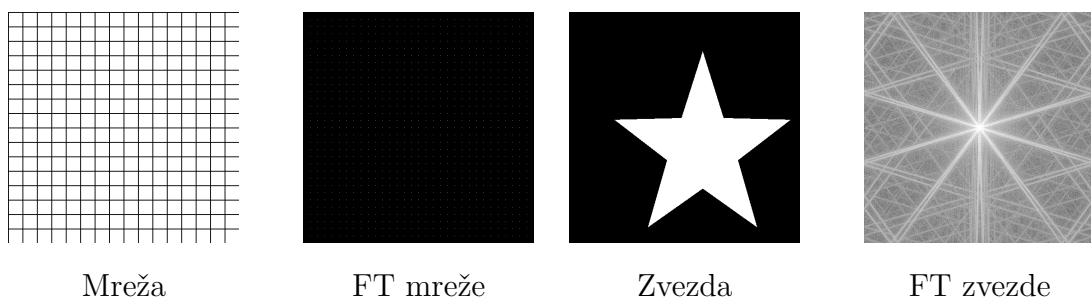
Na spodnjih slikah so prikazane slike Fourierjevih transformacij (FT) slik z osnovnimi geometrijskimi objekti.



Slika 64: Fourierjeve transformacije kroga



Slika 65: Fourierjevi transformaciji pravokotnikov



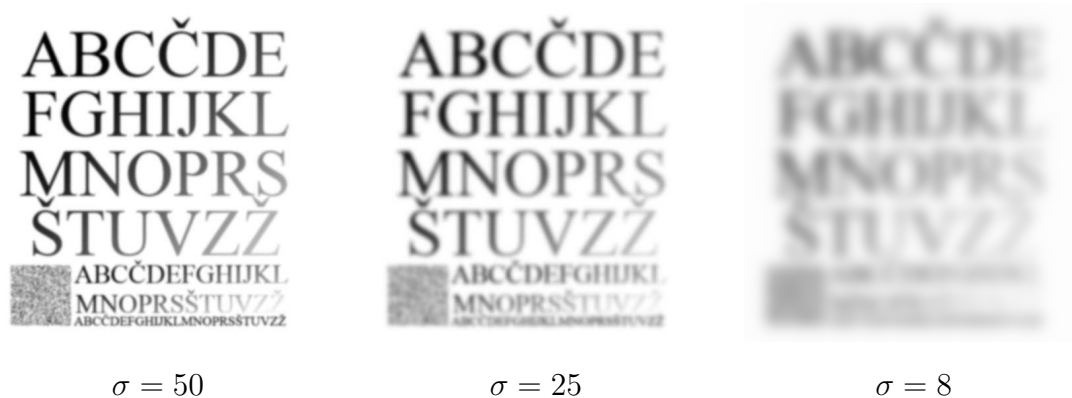
Slika 66: Fourierjevi transformaciji mreže in zvezde

Nizkopasovni filter Z idealnim in Gaussovim nizkopasovnim filtrom sem obdelal sliko z abecedo. Pri uporabi idealnega filtra je lepo viden nihajoči odziv (*ringing effect*), ki pa se pri Gaussovem filtru ne pojavlja.

S zmanjševanjem radija d_0 pri idealnem oziroma variance σ pri Gaussovem filtru se povečuje zamegljenost slike.



Slika 67: Idealni nizkopasovni filter na sliki abecede



Slika 68: Gaussov nizkopasovni filter na sliki abecede

Visokopasovni filter Z visokopasovnim filtrom ohranimo na sliki samo visoke frekvence, torej podrobnosti in robove objektov. Rezultat filtriranja z visokopasovnim filtrom je zelo podoben konvoluciji s katerim izmed operatorjev za iskanje robov (Laplaceov, Sobelov, Prewittov).

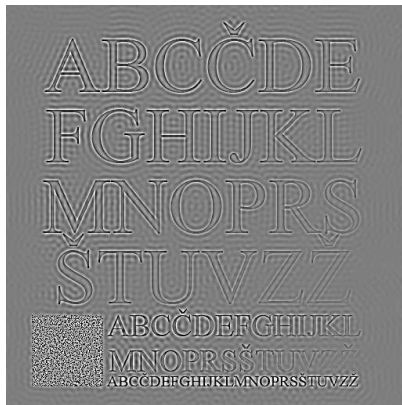


$r = 100$

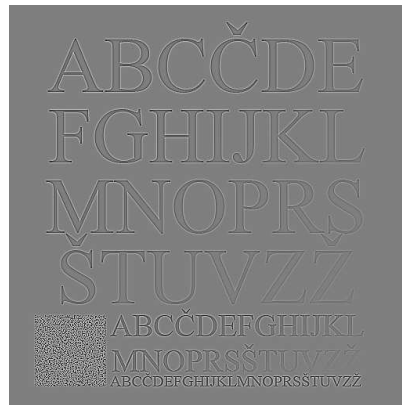
$r = 50$

$r = 20$

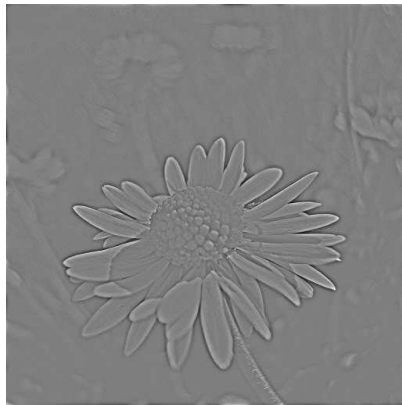
Slika 69: Idealni visokopasovni filter na sliki abecede



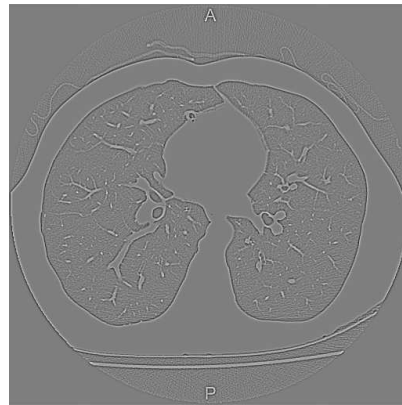
Idealni, $r = 65$



Gaussov, $\sigma = 100$



Gaussov, $\sigma = 35$



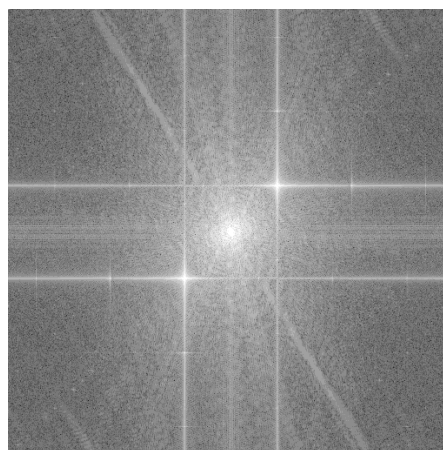
Gaussov, $\sigma = 70$

Slika 70: Centrirani visokopasovni filtri

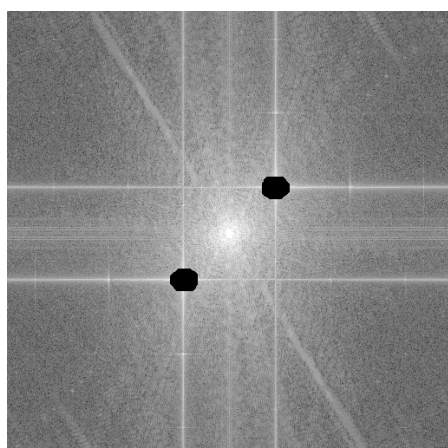
Mešani primeri Slika marjetice je uničena s poševnim vzorcem, ki se ga želimo znebiti. S filtriranjem v prostorski domeni je tak izziv skoraj nemogoče opraviti, medtem ko je filtriranje v frekvenčni domeni namenjeno prav tovrstnim težavam.



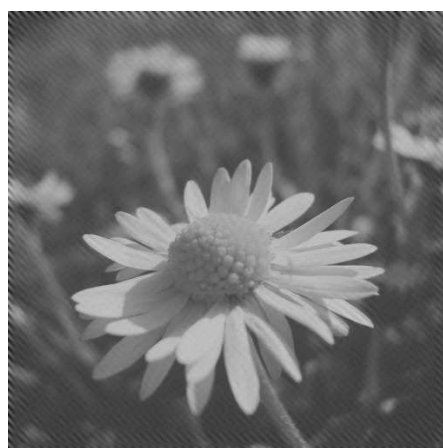
Izvirna slika



FT



Filter z risanjem



Popravljen slika

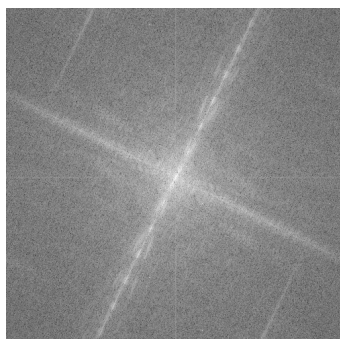
Slika 71: Slika marjetice (L), poškodovane s sinusnim šumom (levo-zgoraj), njena Fourierjeva transformacija, kjer je vidna prisotnost frekvence šuma (desno-zgoraj), z risanjem filtrirana Fourierjeva transformacija (levo-spodaj), popravljen slika marjetice (desno-spodaj)

Primer praktične večstopenjske obdelave sem prikazal na obdelavi slike partiture, iz katere želimo odstraniti notno črtovje (5 vzporednih črt, na katerih ležijo note). Slika je dodatno uničena z dvojnimi sinusnim šumom, zato sem v prvem koraku s frekvenčnim filtrom z risanjem odstranil ta vzorec. Na sliki Fourierjeve transformacije sem počrnil štiri drobne svetle pike, ki nakazujejo prisotnost než-

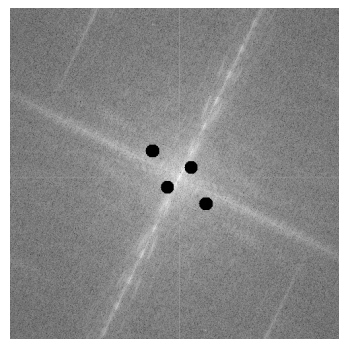
lenega sinusnega šuma. V drugem koraku sem s frekvenčnim filtriranjem s črto odstranil notno črtovje, nato izvedel razteg kontrasta in nazadnje še upragovanje.



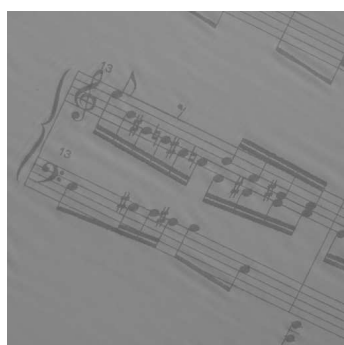
Izvirna slika



FT



Filter z risanjem



Slika po frekvenčnem filtriranju



Slika po filtriranju s črto,



Razteg kontrasta, $\mu = 88$, $\alpha = 26^\circ$, $d = 11$, $\sigma = 15$ $\xi = 4, 18$



Upragovanje, $\mu = 225$

Slika 72: Odstranitev notnega črtovja iz poškodovane slike partiture

Izvirne barvne fotografije Spodnje slike prikazujejo barvne, neobdelane fotografije, ki sem jih zajel z digitalnim fotoaparatom (žaba, marjetica in mesec) ali svetlobnim mikroskopom (krvni razmaz) in uporabil pri primerih obdelave.



Žaba

Marjetica

Krvni razmaz

Mesec

Slika 73: Izvirne barvne fotografije

4 ZAKLJUČEK

V raziskovalni nalogi obdelave digitalnih slik sem se posvetil področju filtriranja in izboljšav, pri katerem sem opisal intenzitetne transformacije in prostorsko ter frekvenčno filtriranje z uporabo konvolucijskih jeder in frekvenčnih filtrov. Z računalniškim programom, izdelanim v razvojnem okolju *Microsoft Visual Studio 2012*, sem kot primer rabe opisanih metod na digitalnih slikah izvedel različne obdelave ter nakazal njihovo aplikativnost na različnih področjih. Program sem objavil tudi na spletu in je prosto dostopen na strani: <http://d-vau.com/rn/>

Najpomembnejša izboljšava programa bi bila možnost obdelave barvnih slik, zaradi razširjenosti visokoločljivostnih digitalnih slik pa bi potreboval tudi dodatno optimizacijo procedur za hitrejšo izvajanje računskih algoritmov.

5 DRUŽBENA ODGOVORNOST

Zaradi vpletenosti digitalnih slik v življenje človeka 21. stoletja je vsekakor pomembno poznavanje osnovne terminologije in principov, ki veljajo na tem področju. Šele s tem znanjem lahko posameznik začne razvijati sistematični dvom ob vsakodnevnem zaznavanju pogostokrat zavajajočih vizualnih informacij iz prenasičenega, k potrošnji usmerjenega okolja in si postaviti prenekatera vprašanja o etičnosti tovrstnega početja. Razvijajoči se mladostniki – najobčutljivejša in najdojemljivejša skupina – se lahko tako z ustreznim izobraževanjem izognejo zlorabam in s trezno sodbo dojemajo lepotne ideale, ki nam jih postavlja oglaševalska industrija.

6 VIRI IN LITERATURA

Literatura

- [1] GONZALES, R. C., WOODS, R. E. 2008. *Digital Image Processing, Third Edition*. New Jersey: Prentice Hall, Inc.
- [2] NERI, E. [et al.] 2008. *Image processing in radiology : current applications*. Berlin: Springer.
- [3] ORLOFF, J. 1997. *Handbook of Charged Particle Optics*. London: CRC Press.
- [4] OSGOOD, B. 2007. *Lecture Notes for EE 261, The Fourier Transform and its Applications*. [Online]. California: Stanford University, Electrical Engineering Department. Dostopno na naslovu:
<http://see.stanford.edu/materials/lsoftae261/book-fall-07.pdf>
[28. 11. 2014]
- [5] PARKER, J. R. 2011. *Algorithms for Image Processing and Computer Vision, Second Edition*. Indianapolis: Wiley Publishing, Inc.
- [6] PAVLIČ, G. 2013. *Spatium Novum*. Ljubljana: Modrijan.
- [7] PETEK, T. 2014. *Izbrana poglavja iz tehniške matematike*. [Online]. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko UM. Dostopno na naslovu:
<http://www.mp.feri.uni-mb.si/osebne/petek/ucbeniki/ucbenikDE.pdf>
[5. 12. 2014]
- [8] STOKES, M. [et al.] 1996. *A Standard Default Color Space for the Internet - sRGB*. [Online]. Dostopno na naslovu:
<http://www.w3.org/Graphics/Color/sRGB> [15. 1. 2014]
- [9] TASIČ, F. J. 2002. *Postopki digitalne obdelave signalov*. [Online]. Ljubljana: F. Dostopno na naslovih:
http://studentski.net/get/ulj_fel_ae1_dos_sno_postopki_digitalne_obdelave_signalov__knjiga_01.pdf
http://studentski.net/get/ulj_fel_ae1_dos_sno_postopki_digitalne_obdelave_signalov__knjiga_02.pdf [5. 12. 2014]
- [10] PLETERŠEK, P. Ž. 2014. *Matematika III*. [Online]. Maribor: Fakulteta za kemijo in kemijsko tehnologijo UM. Dostopno na naslovu:

http://www.fkkt.uni-mb.si/ukemat/Ucbenik_MatematikaIII.pdf [2. 12. 2014]

Slikovno gradivo

- [11] ARSO. [Online]. Dostopno na URL naslovu: http://meteo.arso.gov.si/uploads/probase/www/observ/satellite/msg_ir_sateu_latest.jpg [29. 11. 2014]
- [12] KANDASAMY. [Online]. S pisnim dovoljenjem avtorja. Dostopno na URL naslovu: https://lh5.googleusercontent.com/-NWUksb_9coY/UAQb1HoQVMI/AAAAAAAAA04/d2g-bA0QejA/Kandu6b.jpg?imgmax=912 [15. 12. 2014]
- [13] NASA. [Online]. Dostopno na URL naslovu: <http://imgsrc.hubblesite.org/hu/db/images/hs-2003-28-a-print.jpg> [29. 12. 2014]