UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

ZAKLJUČNA NALOGA

SUPER RESOLUCIJA

OLIVER GLAVINA

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Super Resolucija

(Super Resolution)

Ime in priimek: Oliver Glavina
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Peter Rogelj

Koper, januar 2014

Ključna dokumentacijska informacija

Ime in PRIIMEK: Oliver GLAVINA

Naslov zaključne naloge: Super resolucija

Kraj: Koper

Leto: 2014

Število listov: 33 Število slik: 6 Število tabel: 2

Število prilog: Število strani prilog: Število referenc:

Mentor: doc. dr. Peter Rogelj

Somentor:

UDK:

Ključne besede: super resolucija, obnovitev slik, SR obnovitev

Math. Subj. Class. (2010):

Izvleček:

Super resolucijo opisujejo vsi postopki, ki iz slik nizke ločljivosti pridobijo eno ali več slik visoke ločljivosti.
Postopke super resolucije uporabimo, kadar ni smiselno nadgrajevati senzorjev, ali če želimo zmanjšati ceno strojne opreme.
Super resolucija je v uporabi v medicinskih in varnostnih ustanovah, kjer večja resolucija iz slik nizke ločljivosti pomeni natančnejšo diagnozo ali preiskavo.
V diplomski nalogi se bom osredotočil na definicijo super resolucije in njenega modela. Predstavil bom več metod za izvedbo super resolucije in razlike med njimi. Nazadnje bom prikazal svojo implementacijo metode super resolucije, za katero velja, da so vse slike na primeru posnete z istim senzorjem, in da je gibanje linearno. Delovanje postopka je prikazano na praktičnih primerih.

Key words documentation

Name and SURNAME: Oliver GLAVINA

Title of final project paper: Super Resolution

Place: Koper

Year: 2014

Number of pages: 33 Number of figures: 6 Number of tables: 2

Number of appendices: Number of appendix pages: Number of references:

Mentor: doc. dr. Peter Rogelj

Co-Mentor:

UDC:

Keywords: super resolution, image restoration, SR restoration

Math. Subj. Class. (2010):

Abstract:

Super resolution is defined by all methods that on a set of low resolution LR pictures, which they shift, upscale deblur and merge into one or more high resolution HR pictures.
Super resolution methods are used when upgrading sensors in not feasible or we want to use cheaper hardware.
Super resolution is in use at medical and security facilities, where higher resolution means a more accurate diagnosis or search.
In this thesis I will focus on the definition of super resolution and its model.
I will present multiple methods for implementing super resolution and their differences.
Finally i will show my implementation of a superresolution method, where the pictures from the example are all taken from the same camera and the motion is translational. On the resulting pictures there will be visible improvements on part of the image.

Zahvala

Zahvaljujem se mentorju, dr. Petru Roglju za usmerjanje in nasvete pri izdelavi diplomskega dela in svojemu očetu Ediju za pomoč pri slikanju slik, ki sem jih uporabil v svojih testih.

KAZALO VSEBINE

KAZALO SLIK

KAZALO TABEL

SEZNAM UPORABLJENIH KRATIC

HR	High Resolution
LR	Low Resolution
SR	Super Resolution
CMOS	Complementary Metal - Oxide - Semiconductor
ROI	Region of Interest
PSF	Point spread function
LS	Least Squares
MAP	Maximum a Posteriori
POCS	Projection onto convex sets
IBP	Iterative back-projecting

1 UVOD

Slike so preslikave podatkov iz realnega prizora, ki so bile zajete preko senzorja in shranjene v obliko, ki jo lahko nato uporabimo za njen prikaz. Sliko lahko prikažemo preko tiska, kjer pobarvamo točke na papirju z barvo, ki je določena v zapisu, ali jo prikažemo digitalno, kjer se obarvajo točke na zaslonu. Tem točkam pravimo piksli in so sestavni element vsake slike. Če vsak piksel predstavlja točko na sliki, potem si lahko prestavljamo sliko kot mrežo, sestavljeno iz njih. Velikost take mreže je določena glede na število pikslov, ki jih vsebuje slika in ji pravimo ločljivost slike. To definiramo kot število vrstic in stolpcev mreže na sliki.
Vsaka slika je zajeta z nekakšnim namenom. Če je namen slike zajeti skupinsko sliko, lahko definiramo problem kot:

- zajeti vse ljudi v skupini;
- obraz vsake osebe na sliki mora biti prepoznaven.

Če hočemo zajeti več ljudi v skupini, mora biti slika posneta dlje od oseb, če seveda uporabljamo enake nastavitve optike, in to na isti slikovni napravi. To pomeni, da zgubimo na podrobnostih, ker ima slikovna naprava omejeno število pikslov in je pri večjem prizoru ločljivost manjša glede na število pikslov na cm. Če pride do takega primera, pomeni, da senzor, s katerim poskušamo zajeti sliko, ni primeren in zajete slike so slike nizke ločljivosti (LR). Sliko nizke ločljivosti (LR) definiramo kot sliko, katere resolucija ne ustreza našim zahtevam. Želimo dobiti sliko, ki je ustrezna. Takšni sliki pravimo slika visoke ločljivosti (HR).
Potreba po HR slikah se pokaže na veliko področjih. V medicini gostota pikslov pomeni večjo natančnost pri določitvi diagnoze, kar posebej velja pri uporabi magnetne resonance, kjer smo omejeni na LR slike. HR slike so zaželene tudi na področju varnostnega nadzora in forenzične preiskave, kjer želimo povečati del slike ROI (region of interest), ki predstavlja področje slike, ki vsebuje največ podatkov za reševanje zastavljenega problema. HR slike potrebujemo tudi pri računalniškem vidu, saj če slika ne zadostuje zadanemu problemu, program za računalniški vid lahko napačno oceni podatke in ne dobimo zaželenega rezultata.
Tudi v primeru napovedi vremena potrebujemo HR satelitske slike, ki pripomorejo k natančnosti ocene vremena.

Poznamo več možnih rešitev za pridobivanje HR slik. Ena rešitev je zamenjava senzorja, s katerim zajamemo slike, z boljšim. Pri načrtovanju novega senzorja se lahko odločimo, da zmanjšamo velikost, ki ga zavzame posamezni piksel na senzorju. Tako zagotovimo večjo gostoto pikslov na senzorju in tudi večjo ločljivost slik. Takšna rešitev je omejena z difrakcijskim limitom. Manjša je velikost posameznega piksla, manj svetlobe pade na del senzorja, ki pripada takemu pikslu. Zaradi manjše količine svetlobe, ki jo lahko zaznamo na senzorju, le-ta pomeni manjšo količino koristnega signala in je razmerje signal / šum manjše.

To se odraža v bolj ″pošumljeni″ sliki. Izvemo, da je tako optimalna velikost piksla 40 μm2 za 35 μm CMOS postopek [1]. Drugi možni način za izboljšavo senzorja je povečati njegovo velikost. To je povečava velikosti celotnega senzorja, kjer imamo večjo mrežo pikslov. Povečava velikosti senzorja velja za drago rešitev, saj potrebujemo tudi optiko, ki pokriva vso površino senzorja. Z velikostjo senzorja naraste tudi velikost objektiva in zato tudi cena opreme.

Razvoj senzorjev za znanstvene namene je drag, saj stane več milijonov dolarjev. Večje izboljšave na senzorju prinašajo zmeraj manj pridobitev na kvaliteti slik glede na vloženo vsoto denarja. Tako pridemo do meje, kjer se ne splača več vlagati v strojno opremo in moramo iskati še kakšno rešitev drugod, saj želimo tudi splošnim uporabnikom kamer in videokamer zagotoviti kvalitetnejši zajem slik. Rešitev se nam ponudi v obliki programske opreme, kjer iz več zajetih LR slik lahko dobimo eno ali več HR slik. Takšnemu postopku pravimo super resolucija oz. SR. Največja prednost pri tej tehniki je cena, saj zadostuje že poceni oprema, ki lahko zajema LR slike, ki jih potem obdelamo s tehniko super resolucije. Druga prednost pa je ohranitev trenutne opreme za uporabnike, ki že imajo naprave za zajemanja slik. Tako ni potrebno uvajati nove opreme v delovne prostore oz. nadgradnja strojne opreme uporabnika.

Za tehnike SR obstajajo tudi določene omejitve, kjer je rekonstrukcija HR slike uporabna le z več LR slik istega prizora, za katere zahtevamo, da je prizor razmeroma statičen.

Delovanje SR metod lahko opišemo kot postopek, ki kot vhodni podatek uporabi množico LR slik istega prizora. Vsaka slika v LR množici se razlikuje med seboj na podtočkovni ravni. Ker se slike razlikujejo na podtočkovni ravni, in ker za vsako sliko velja, da je ni mogoče izpeljati iz drugih LR slik iz iste množice, velja, da vsaka slika prikazuje edinstven pogled na prizor.

Ob zajemanju LR slik pride do naravne izgube podatkov prizora: zameglitve zaradi omejene hitrosti zaklopa kamere, optičnega popačenja slike (difrakcijskega limita), šuma in omejene gostote senzorja. Ker so LR slike izpostavljene takšni izgubi, je naloga SR metod tudi obnovitev slik iz zamegljenih, nečistih slik. Ker nam vsaka LR slika predstavlja drugačno sliko prizora, lahko izkoristimo in združimo vse podatke o prizoru, ki veljajo za različne slike iz LR množice in iz njih tvorimo HR sliko, ki zadošča rešitvi našega problema.

Namen diplomskega dela je predstavitev tehnike super resolucije, prikazati že razvite algoritme za izvedbo super resolucije in implementacijo SR algoritma ter preizkus njegovega delovanja na zajeti množici LR slik. Diplomsko delo bo razdeljeno na več delov. V prvemu delu bom predstavil problem super resolucije in opazovalne modele. Drugi del naloge bom namenil predstavitvi že raziskanih algoritmov za super resolucijo. Tretji del diplomskega dela bo namenjen preverbi algoritma super resolucije na primeru. Predstavil bom delovanje izbranega algoritma, metode in podatke, ki sem jih uporabil pri izvedbi preizkusa ter predstavil nastale težave, ugotovitve in rezultate.

2 PROBLEM SUPER RESOLUCIJE

2.1 Definicija rekurzivnega problema

Problem super resolucije si lahko zastavimo kot pridobitev HR slike, ki je bila popačena zaradi omejitev optičnega sistema. Takšnemu problemu rečemo inverzni problem, saj na LR slikah hočemo izvesti obratne operacije od tistih, ki so se izvedle ob zajetju slike (popačenja) in oceniti najboljši približek originalni HR sliki. Glede na naše vhodne podatke in modele šuma lahko izberemo takšno metodo, ki nas bo pripeljala do najboljše ocene HR slike. Takšnim metodam pravimo optimalne metode, ki so modelirane na pričakovanih vhodnih podatkih. Če se v vhodnih podatkih pojavi en nepredvideni podatek, za katerega funkcija ni bila ustrezno modelirana, se ocena HR slike pokvari. Zaradi tega razloga se izogibamo uporabi optimalnih metod, saj moramo zagotoviti tudi točnost podatkov. Le eden nepredviden podatek popači končni rezultat. Rajši uporabljamo neoptimalne metode, za katere velja, da so bolj stabilne, saj niso toliko občutljive na tip vhodnih podatkov in model šuma. Trdnost SR algoritmov in učinek nepredvidenih podatkov na vhodnih slikah lahko izmerimo s točko razpada. Ta nam natančno določi, koliko nepredvidenih podatkov je potrebnih, da se nam končna ocena HR slike pokvari. Vzemimo za primer enostavnejše metode ocene, kot je ocena povprečja, ki naredi povprečje vseh slik iz LR množice slik in iz tega oceni HR sliko. Ocena povprečja ima točko razpada 0 in je potreben le en podatek za negativen vpliv na končno oceno slike. Trdnejša metoda ocene je mediana, ki oceni vsak piksel v množici LR slik glede na njihove srednje vrednosti. Točka razpada za mediano je 0,5, ker je potrebno več kot 50 % nepredvidenih elementov v vhodnih podatkih, da se ocena HR slike pokvari.
Za nadaljevanje si oglejmo opazovalni model zajemanja slik. Le-ta nam pokaže vse operacije, ki nas pripeljejo do LR slike.

2.2 Opazovalni model

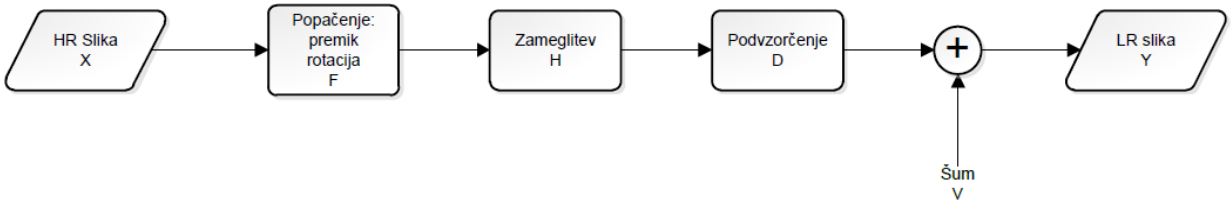
Opazovalni model za super resolucijo, kjer je predpostavljeno linearno (translational) gibanje, so obravnavali v člankih [1-6]

$$\underline{Y_k} = \underline{M_k}\underline{X} + \underline{V_k}, \quad (1)$$

Kjer $\underline{Y_k}$ predstavlja popačeno LR sliko (oz slike), $\underline{M_k}$ je sistem, s katerim smo sliko zajeli, $\underline{X}$ je originalna HR slika scene, preden jo sistem popači, $\underline{V_k}$ je naključni šum, ki je prisoten pri katerem koli sistemu zajemanja slik in k predstavlja zaporedno številko v zaporedju zajemanja slike razvrstitvi zajetih slik. Predstavljeni zapis je v vektorskem formatu.
Opazovalni model zajema slik prikaže vse operacije, vključene v izdelavo LR slike, z izbranim sistemom. Opazovalni model nam pomaga pri rešitvi problema SR obdelave LR slik, saj je SR obdelava inverzni problem, zato moramo obrniti operacije v opazovalnem modelu, da dobimo želeno HR sliko. Operacije, ki HR sliko popačijo, so vključene v sistem zajemanja slik $\underline{M_k}$. Za boljše razumevanje operacij vpliv sistema $\underline{M_k}$ lahko razbijemo na manjše dele:

$$\underline{M_k} = D_k H_k F_k, \quad (2)$$

Kjer D_k predstavlja učinek spremembe podatkov iz višje na nižjo resolucijo (podvzorčenje ali downsampling), H_k predstavlja učinek zameglitve na optičnem sistemu ob zajetju slike in F_k predstavlja geometrijsko popačenje slike, kot je na primer rotacija ali premiki. Vrednost k predstavlja k-to sliko zajeto v zaporedju iz množice zajetih LR slik. Če so vse slike iz LR množice zajete preko istega optičnega sistema, velja, da uporabimo enak PSF(Point Spread Function) za vse zajete slike. PSF je razpršilna funkcija, ki določa razpršitev pikslov in je zato učinek zameglitve H_k enak za vse slike in ga označimo z H. Enako velja za operacijo podvzorčenja D_k, kjer pri uporabi istega optičnega sistema in zajemu slik pri enaki resoluciji velja, da je učinek enak za vse slike in ga označimo z D.



Slika 1: Kako iz HR slike originalne scene pridobimo popačeno LR sliko z dodanim šumom

$\underline{M}_k$ vsebuje podatke o globalnih ali lokalnih premikih na prizoru, rotacijo in druge vrste transformacij na sliki. Podatke o transformaciji označimo z F_k kot učinek geometrijskega popačenja slike. Informacija o transformaciji je neznana in jo moramo predvideti za vsako sliko iz LR množice slik posebej. Postopek popačenja na $\underline{X}_k$ je definiran glede na razdaljo premikov v pikslih. Gibanje za vsako sliko lahko predvidimo s posebnimi algoritmi za detekcijo premika. Če je gibanje enostavno, ocenimo gibanje za vsako sliko po stolpcih in vrsticah glede na začetek in konec gibanja. Ker mora biti ocena natančna na podtočkovni ravni, se morajo piksli prilagoditi točkovni mreži z uporabo interpolacije pikslov.To je ocena neznane vrednosti piksla glede na znane podatke, v našem primeru podtočkovne. Med najbolj uporabljenimi interpolacijami je bilinearna interpolacija, kjer se izvede linearna interpolacija v dveh smereh, za vrednost stolpca in vrstice na mreži točk.

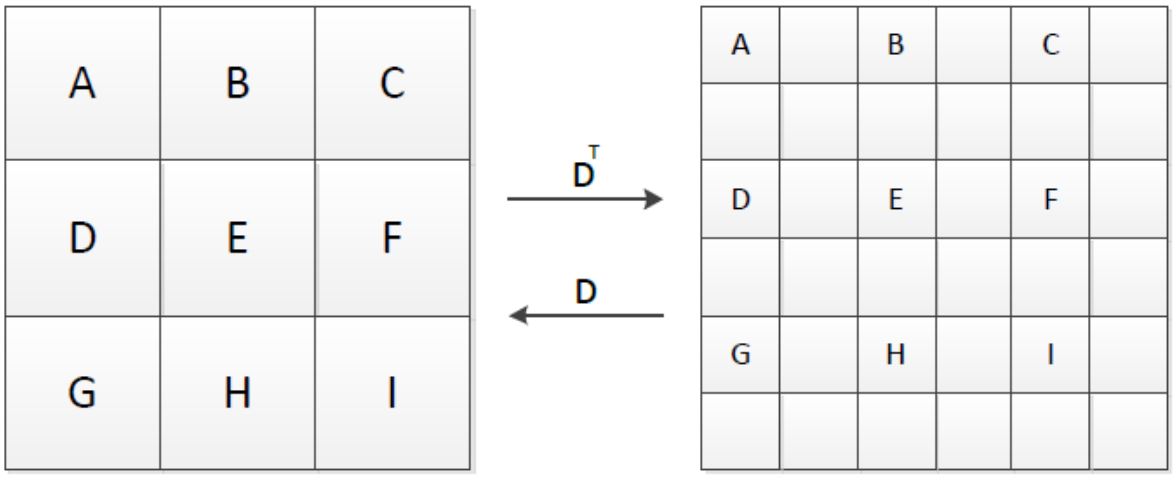
Zameglitev povzroči optični sistem (izven fokusa, difrakcijski limit) s premikom med optičnim sistemom in originalnim prizorom in razpršilno funkcijo PSF optičnega senzorja. V linearnem prostoru, kjer je optični sistem isti, je zameglitev invariantna in jo označimo z H. Ta lastnost nam dovoli uporabo enostavnih algoritmov za ocenitev zameglitve H. Oceno zameglitve H lahko modeliramo kot PSF optičnega senzorja po operatorju povprečenja na LR sliki, ali pa uporabimo približek funkcije zameglitve LR slike (Gaussian Blur).

Operacija podvzorčenja D generira slike nižje resolucije od originalnega prizora. Ker je število pikslov LR slike določeno z resolucijo senzorja, so podatki za vsak piksel LR slike rezultat povprečja vrednosti, ki so padle v tisti piksel. Zaradi tega pride pri podvzorčenju do popačenja slike in ostrih robov. Če hočemo obrniti učinek podvzorčenja, moramo povečati resolucijo in preslikati vse piksle na točkovno mrežo HR slike. Po preslikavi na HR mrežo (slika 2) so piksli razpršeni na HR mreži z neznanimi vmesnimi vrednostmi pikslov, ki so nastali zaradi povečave resolucije slike. Vmesne vrednosti dobimo z interpolacijo med piksli, kar povzroči dodatno zameglitev pri oceni SR slike.

Iz opazovalnega modela v (1) lahko razdelimo postopek SR obnovitve slike na 3 korake: registracijo, interpolacijo in obnovitev slike. Glede na implementacijo lahko izvedemo vse 3 korake istočasno, ali pa se vsak izvede posebej. V fazi registracije ocenimo premik prizora na

vsaki LR sliki posebej. Glede na to, ali je gibanje globalno ali lokalno, linearno ali kompleksno, uporabimo drugačne metode za oceno gibanja. Gibanja moramo oceniti z veliko natančnostjo, saj je lahko premik manjši od velikosti piksla in natančna ocena gibanja igra zelo pomembno vlogo za rekonstrukcijo HR slike. V drugi fazi moramo piksle preslikati na HR mrežo in opraviti interpolacijo med piksli za izračun vmesnih vrednosti. Zadnja faza je obnovitev slike, kjer se znebimo šuma in zameglitve, ki smo ju pridobili po interpolaciji pisklov zaradi transformacije gibanja in nadvzorčenja slik in znebiti se moramo zameglitve, ki je nastala zaradi PSF optičnega senzorja.

Poleg modela, ki sem ga predstavil v (1) in na katerega sem se osredotočil v tem diplomskem delu, obstajajo še drugačni modeli, ki ustrezajo drugačnim vhodnim podatkom ali sistemu. Eden od takih modelov je model super resolucije z upoštevanjem afine transformacije. Model je bil ustvarjen za reševanje problema super resolucije v primerih, ko se objekt premika ali oddaljuje od kamere, torej se na sliki spreminja velikost objekta, ali ko je slika obrnjena za določen kot, kjer moramo spraviti obrnjene piksle LR slik na ne obrnjeno HR mrežo. Takih primerov ne moremo opisati s predstavljenim enostavnim SR modelom.



Slika 2: Prikaz funkcije podvzorčenja in njene obratne funkcije, kjer preslikamo piksle A, B , C,… na HR mrežo in moramo vmesne vrednosti med piksli določiti z interpolacijo.

3 SR ALGORITMI

Učinkovitost SR algoritma se pokaže v času izvajanja postopka in v zahtevi po strojni opremi. Problem, na katerega naletimo pri izvajanju super resolucije, je število možnih rešitev, ki jih le-ta lahko vrne kot rezultat za določeno množico LR slik. Zaradi razsežnosti problema je smiselna vključitev iterativnega postopka v SR algoritem, saj razbije problem na manjše enote, ki jih lažje obdelamo in združimo v končno HR oceno. Izbrati hočemo torej algoritem, ki je računsko čim manj zahteven in zagotovi sprejemljive rezultate kljub nepričakovanim vrednostim v podatkih. V tem poglavju sem vzel za zgled članka [3] in [4], ki opisujeta tipe SR algoritmov in njihove prednosti ter slabosti.
Optimizacija je najbolj razvidna na operacijah transformacij optičnega sistema M in njihovih obratnih operacijah. Sledeči algoritmi sledijo modelu SR, prikazanem v (1), za katere bomo privzeli linearno gibanje brez rotacij v ravni črti.
SR algoritme delimo na postopke, ki spadajo v frekvenčno in prostorsko domeno. V primeru frekvenčnih metod primerjamo razmerje LR slik z želeno HR sliko za dano transformacijo gibanja. Za SR model s frekvenčno metodo je predviden linearni globalni premik (možno ga je prilagoditi tudi ostalim transformacijam) z invariantnim šumom na slikah, kjer za njihovo rekonstrukcijo uporabimo enačbo z lastnostmi Fourierjeve transformacije DFT LR slik glede na zvezno transformacijo želene HR slike, kjer je pasovna širina signala slike omejena. V diplomskem delu sem se osredotočil na algoritme v prostorski domeni, ki niso omejeni na globalno transformacijo, ampak so modelirani po opazovalnem modelu iz (1). Za razliko od frekvenčne domene lahko v prostorski domeni obdelamo slike, katerih gibanje je linearno ali nelinearno, je lahko globalno ali lokalno in upoštevamo zameglitve premika in sistema, podvzorčenja in šuma na sliki. V prostorski domeni lažje uporabimo tudi apriori parametre, s katerimi omejimo število rešitev za dani SR problem.

3.1 Ne-uniformna interpolacija

Ne-uniformna interpolacija je enostaven postopek, kjer so operacije SR (registracija, interpolacija in obnovitev slike) izvedene po korakih ena za drugo. Piksle LR slik preslikamo na HR mrežo in izvedemo ne-uniformno interpolacijo direktno ali iterativno. Po končani interpolaciji odstranimo nastalo zameglitev, s katero koli metodo, ki upošteva šum. Ne-uniformna interpolacija je računsko nezahteven algoritem in je zmožen obdelovanja LR slik v realnem času, kjer lahko opazujemo učinek metode SR istočasno z izvajanjem programa. Metode interpolacije so enostavni postopki, ki ne zmorejo pridobiti višjih frekvenčnih komponent prizora, kot so prisotni na LR slikah. Z metodo tudi privzamemo, da so vse značilnosti zamegljenosti in šuma enake na vseh slikah, kar povzroči nepravilnosti na končni oceni, če temu ni tako.

3.2 Iterativna obnovitev z vzvratno projekcijo

Pri iterativni SR obnovitvi z vzvratno projekcijo (IBP) je HR slika ocenjena z razliko med množico zajetih LR slik in množico simuliranih LR slik. Ta postopek je ponovljen iterativno za vsak korak tako, da minimizira energijo napake. Množico simuliranih LR slik pridobimo iz trenutne HR ocene slike na iterativnem koraku. Dobljeno razliko pomnožimo z jedrom vzvratne projekcije, katero prištejemo HR oceni slike iz njenega modela. Če obstaja rešitev za razlike med zajetimi LR slikami in simuliranimi LR slikami, nam jedro projekcije določa faktor vpliva operacije projekcije na oceno HR slike. Algoritem IBP je za predstavitev zelo enostaven, ampak obstaja problem pri določitvi primernega jedra vzvratne projekcije in pri vključitvi vnaprej določene vrednosti v postopek.

3.3 Regulariziran pristop SR obnovitve

Obnovitev SR slik s SR metodo je problem, ki zajema veliko število možnih rešitev zaradi neznanih operatorjev zameglitve in števila vhodnih LR slik. Z večjo količino LR slik lahko zmanjšamo problem, ampak če jih imamo v omejeni količini, moramo uporabiti postopke regularizacije, ki omejijo število možnih rešitev pri pomanjkanju podatkov.
Postopke regularizacije delimo na deterministične in stohastične, kjer pri deterministični metodi uporabljamo least-squares metodo (CLS) in maximum a posteriori (MAP) metodo regularizacije pri stohastičnih postopkih.

Za deterministični pristop regularizacije SR obnovitve lahko definiramo registracijske parametre z vnaprej določenimi vrednostmi in uporabimo least-squares metodo, za katero velja, da iščemo rešitev, za katero je vsota napak končne ocene najmanjša. Kot vnaprej določeni podatek o želeni rešitvi pri determinističnem pristopu uporabimo parameter za omejitve glajenja slike. Ker so LR slike podvzorčene z ostrimi robovi, so slike naravno brez ostrih robov. Zaradi tega v postopek obnovitve implementiramo filter, namenjen glajenju ostrih robov na slikah. Učinek filtra z vnaprej določenim znanjem omejimo za našo želeno HR sliko z omejitvijo glajenja slike. Parameter omejitev glajenja slik je vnaprej določena vrednost, ki jo definiramo glede na želeno količino glajenja robov in na število LR vhodnih slik, kjer je pri manjši količini priporočena večja vrednost parametra glajenja.
Za stohastični pristop regularizacije SR obnovitve uporabimo statistično reševanje problema, po navadi Bayesov model, kjer uporabimo skupaj z vnaprej določenimi vrednostmi želene HR slike še maximum a posteriori MAP vrednosti. MAP oceno izračunamo iz vrednosti LR slik glede na vnaprej določene vrednosti o HR sliki in statističnih informacijah o šumu. Izmed možnih rešitev z Bayesovo metodo ocenimo najboljšo rešitev SR problema glede na vnaprej določeni model slik in uporabo MRF (Markov Random Field) polj. Predpostavimo, da so slike naravno gladke kot pri determinističnem pristopu. Potem lahko vključimo v Bayesov postopek

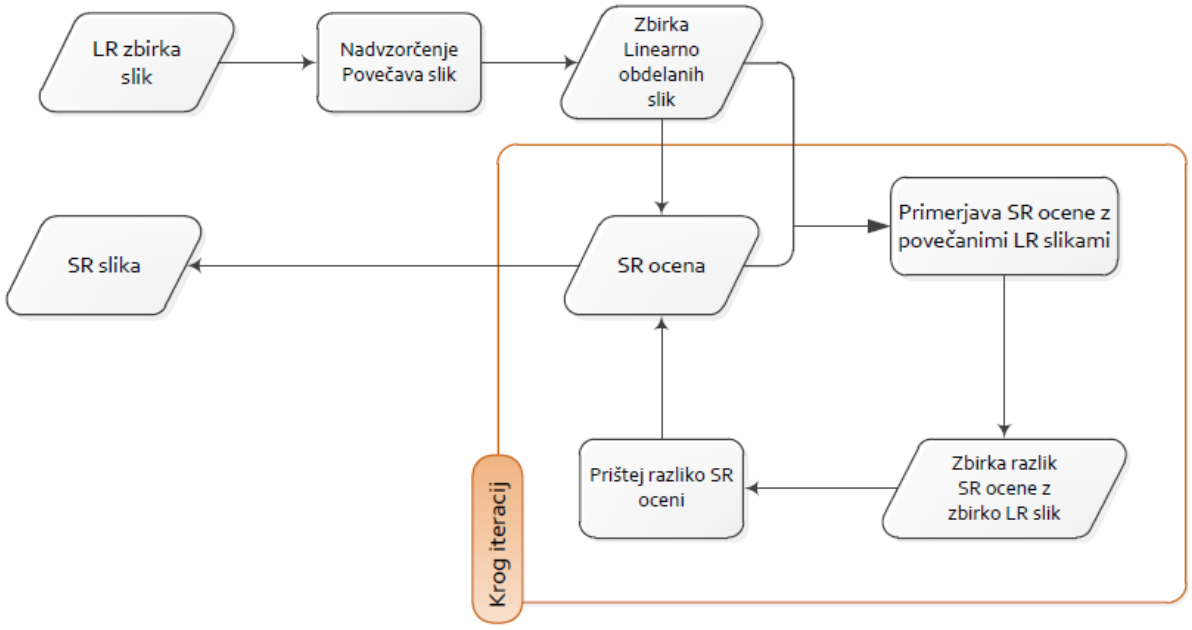
Gaussove vnaprej določene vrednosti, ki omejijo visoke frekvence na HR oceni slike. Z omejitvijo učinka Gaussovih vnaprej določene vrednosti dobimo HR sliko z zglajenimi in ohranjenimi robovi. Poleg MAP obstaja tudi Maximum Likehood (ML) metoda [2], kjer statistično rešujemo SR problem kot pri MAP, ampak brez vnaprej določene vrednosti.

3.4 Projekcija na konveksne množice

Projekcija na konveksne množice (Projection onto convex sets POCS) je vrsta iternativne metode, ki vključuje vnaprej določene podatke o HR sliki v fazi obnovitve. Množico rešitev SR problema omejimo s konveksno množico omejitev, ki predstavljajo želene lastnosti končne SR slike, kot so na primer gladkost, omejitev energije in točnost podatkov. Podatki omejitve so zapisani v konveksni množici v obliki vektorjev, na katere projiciramo prostor rešitve problema SR. Za izbrano točko v prostoru operator projekcije konveksne množice projicira izbrano točko iz množice SR rešitev in jo projicira na ustrezno mesto v konveksni množici. Z nadaljnjimi iterativnimi koraki izbrana točka konvergira k eni sami rešitvi, ki pa ni unikatna in je odvisna od začetne izbire točke v prostoru rešitev SR. Takšen algoritem je sposoben istočasno dokončati fazo interpolacije in obnovitve slike, če lahko dobro ocenimo podatke v fazi registracije. Slabost algoritma je tudi prevelika računska cena in počasna konvergenca rešitve.

4 IMPLEMENTACIJA METODE SUPER RESOLUCIJE

Za izvedbo obnovitve LR slik z uporabo SR metode sem izbral iterativni algoritem z vzvratnim projiciranjem. Izbral sem IBP algoritem, pri katerem sem primerjal SR oceno z vhodnimi slikami za vsak iterativni korak ter rezultat regulariziral, kar je zmanjšalo pridobljeni šum na sliki. Kot glavno prednost sem upošteval enostavnost in intuitivnost algoritma,v katerega sem lahko preslikal funkcije SR iz opazovalnega modela (1) in implementiral metodo SR obnovitve. Ideja implementacije algoritma je razdeliti zastavljeno nalogo na 3 faze: poravnavo, interpolacijo in obnovitev. V prvi fazi bom med registracijo štel tudi pridobitev LR slik, na katerih sem izvajal metodo SR za obnovitev HR slike. LR slike, ki sem jih uporabil v metodi SR, so bile zajete na dva načina. Prvi način zajemanja slik sem preizkusil z implementacijo metode, ki je preko vhodne kamere zajela slike prizora. Vzorec LR slik sem nato uporabil v metodi SR. Drugi način pridobitve slik je zajem posameznih slik, ki sestavljajo predhodno zajeti videoposnetek. V obeh primerih smo predpostavili, da je gibanje linearno. Linearno gibanje opisuje tako vrsto gibanja, ki v zajeti sekvenci slik ne spreminja smeri in ima konstantno hitrost gibanja.



Slika 3: Algoritem za SR obdelavo

Pri implementaciji SR algoritma, prikazanega v sliki 3, sem uporabljal c++ programski jezik, v katerem sem implementiral svoje metode in openCV (Open Source Computer Vision Library) knjižnice. To je odprtokodna knjižnica za računalniški vid in strojno učenje, iz katere sem uporabljal podatkovne strukture za shranjevanje slik in metode za zajem slik.

4.1 OpenCV

V tem delu bom predstavil SR razred iz openCV knjižnice, ki sem jih preizkusil na testnem programu, priloženem v dokumentaciji openCV. Primerjal ga bom z algoritmom iz slike 3 in ugotovil, ali dobimo ustrezne rezultate.
OpenCV ima definiran razred za super resolucijo, ki je zmožen obnoviti LR posnetek v HR posnetek scene. Osnovni razred super resolucije deluje na takšen način, da določimo vhodne podatke (input) s pomočjo metode superres::SuperResolution::setInput(const Ptr<FrameSource> & frameSource). Vhodne podatke sprejema kot video ali pa kamero povezano na sistem. Metoda superres::SuperResolution::nextFrame obdela sliko iz frameSource z načinom SR obnovitve in vrne dobljeni rezultat kot sliko.
Metoda superres::SuperResolution::collectGarbage počisti vse podatke v pomnilniku.
Z metodo superres::createSuperResolution_BTVL1 oziroma z metodo, ki uporablja grafični procesor superres::createSuperResolution_BTVL1_GPU, ustvarimo super resolucijo in ji definiramo lastnosti:

Tabela 1: Lastnosti za konstruktor SR razreda ″superres″

Int scale	Faktor povečave
Int iterations	Število iteracij
Double tau	Vrednost metode SD
Double lambda	Parameter teže za uravnoteženost razmerja med podatki in gladkostjo
Double alpha	Parameter za prostorno distribucijo pri bilateralnem TV
Int btvKernelSize	Velikost jedra B-TV filtra
Int blurKernerSize	Velikost jedra Gaussove zameglitve
Double blurSigma	Sigma za Gaussovo zameglitev
Int temporalAreaRadius	Premer časovnega iskalnega področja
Ptr<DenseOpticalFlowExt> opticalFlow	opticalFlow algoritem za oceno premika na slikah

Preizkus delovanja razreda sem vzel na testnem programu iz dokumentacije openCV. Kot vhodne parametre je program zahteval video posnetek, ki je bil določen kot frameSource za postopek super resolucije. Za frameSource lahko določimo kamero, priključeno na računalnik, ali že vnaprej shranjen videoposnetek. Za frameSource sem nastavil video, ki sem ga predhodno posnel s kamero. Glavna prednost, ki jo prinaša uporaba že vnaprej posnetega video posnetka za frameSource SR postopka je, da omogoča primerjavo rezultatov pri različnih nastavitvah, saj so vhodne LR slike iz video posnetka vedno enake.

Z uporabo metode nextFrame sem dobil vrnjeno SR sliko za vhodno LR sliko, vzeto iz video posnetka, definiranega za frameSource. Metodo nextFrame sem klical za vnaprej določeno število slik, imgNum. To predstavlja število slik od začetka video posnetka in bodo uporabljene v SR obnovitvi. Po vsakem klicu metode nextFrame je bil izveden postopek obnovitve SR za vsako LR sliko in zapis dobljene HR slike v novi video posnetek. Rezultat je bil video posnetek, ki je bil povečan za faktor scale, kjer je bila vsaka slika posnetka obdelana s funkcijo SR.

Cilj implementacije SR metode je pokazati SR način na enostavnem primeru. Želimo, da po SR obdelavi LR slik dobimo eno HR sliko. Primer z uporabo knjižnic openCV ne zadostuje za prikaz SR metode, za katero privzemamo linearno gibanje objekta v sceni.

Tabela 2: Shema programa za enostavno super resolucijo

Ime spremenljivke	Opis
Nastavitve kamere	
takes	Število slik, ki jih bo zajela kamera
captureSpeed	Interval (v milisekundah) med vsako sliko
Nastavitve super resolucije	
scale	Faktor povečave slike
iterations	Število iteracij v iterativni SR metodi
imgNum	Število LR slik, uporabljenih v SR metodi
folder	Mesto LR slik
movex	Globalni premik posamezne slike na x osi
movey	Globalni premik posamezne slike na y osi
ker	Velikost jedra
alpha	Parameter prostorske distribucije
beta	Vrednost velikosti koraka pri ″steepest descent″ minimizaciji
lambda	Vrednost omejitve gladkosti pri regularizaciji napak / odpravljanju šuma

4.2 Implementacija enostavne SR

Program za enostavno super resolucijo sem razdelil na dva dela. V prvem delu sem se osredotočil na zajem slik in pridobitev LR množice slik, katerega potem posredujemo drugemu delu programa, ki se ukvarja s SR obdelavo slik. Drugi del programa sprejme polje LR slik in jih obdela ter združi v eno HR sliko. Pred delovanjem prvega in drugega programa je potrebno definirati vrednosti vhodnih spremenljivk oz. nastavitve za metodo super resolucije:
Tu so takes, captureSpeed, imgNum in folder uporabljene za prvi del programa (v fazi zajemanja slik). Scale, iterations, imgNum, movex, movey, window, alpha, beta in lambda pa so uporabljene v drugem delu programa (v fazi SR obdelave).

4.2.1 Zajem LR slik

V našem primeru bom zajel slike s statično kamero. Ker je kamera statična, je njena prednost v tem, da se lahko osredotočimo na določeni del slike, kjer se izvede linearni premik in pridobimo SR na lokalnem področju. Zaradi natančnosti meritev je tudi lažje zagotoviti statičnost kamere. Če bi se odločil za način, kjer kamero premikamo linearno, je večja verjetnost, da pride do napak v meritvah, kjer se scena ali objekt premakne v isti smeri ali obratni smeri od premikanja kamere. V takem primeru je LR slika zunaj ležeči primer. Če hočemo zagotoviti najboljše opravljeno SR obnovitev slike, moramo zmanjšati, ali odpraviti število zunaj ležečih primerov. Zajem množice LR slik je mogoč tudi z več kamerami, ki slikajo isto sceno z linearnim zamikom, vendar zgubimo na enostavnosti sistema in strojna oprema je dražja.
Za zajem in upravljanje s slikami uporabljam openCV knjižnice. V prvem delu odprem kamero priključeno na računalnik z:

$$Videocapture\ cap(0);$$

ki odpre privzeto kamero. V primeru več priključenih kamer, kjer je na prenosniku že vgrajena kamera in nanj priključimo spletno kamero ali kakšen drugi video vhod, Videocapture cap(0) odpre nazadnje priključeno kamero. Iz video vhoda cap(0) shranjujem slike v podatkovno strukturo Mat. Razred Mat v openCV knjižnici je definiran kot razred 2D polj, ki predstavljajo matriko in je idealen za hranjenje slik. Map objekti so lahko tudi večkanalni (R, G, B) in podpirajo področni dostop do podatkov (region of interest ROI).
Glede na nastavljene spremenljivke takes in captureSpeed izvaja zanko, kjer za spremenljivko takes predstavlja, kolikokrat se bo zanka izvedla in captureSpeed predstavlja, koliko časa bo zanka čakala. V zanki vsakič zajame LR sliko iz video vhoda cap in ga shrani v matriko frame. Sliko v matriki frame nato shrani na privzeto lokacijo, kjer se nahaja program, pod imenom ″inLRX″, kjer je X= 0; X < takes za vsak korak v zanki. Po končanem zajemu dobljene LR slike shrani v Polje matrik Mat: Mat imgs[X]. Odprta je tudi možnost uporabe predhodno zajetih slik, z uporabo spremenljivke folder, ki definira pot datotek LR slik in imgNum, ki določi, koliko slik bo naloženih in uporabljenih v SR obdelavi.

4.2.2 SR obdelava slik

Drugi del programa služi SR obdelavi LR slik. Problem SR obdelave sem razdelil na štiri korake: interpolacijo, registracijo premika, obnovitev in izostritev. Korake izvedemo v zaporedju z izjemo registracije premika, ki ga lahko izpustimo, če predhodno podamo podatek o koraku premika.

Interpolacija slik

Ker je cilj dokazati super resolucijo, mora biti končni rezultat slika z večjo količino pixlov in pridobitvijo na podrobnostih slike. Množico LR slik, ki so spravljene v polju matrik imgs[], povečamo število stolpcev in vrstic LR slik za faktor definiran v spremenljivki scale. Linearno interpolacijo izvedemo na matriki LR slik z metodo resize:

$$resize(frame, imgs[i], s, scale, scale);$$

kjer frame predstavlja pomožno matriko, v kateri se nahaja trenutna LR slika, imgs[i] je ciljni prostor za povečano LR sliko, s je privzeta vrednost velikosti in scale je faktor povečave ločljivosti. Metodo resize izvajamo za vsako sliko v zanki, kjer $\mathrm{i} = 0;\ \mathrm{i} <$ imgNum in izvedemo povečavo pred vsemi drugimi operacijami, ker tako največ pridobimo na informaciji slike. Po končani SR obdelavi pridobimo na točnosti in podrobnostih.

Registracija gibanja

V fazi registracije gibanja je cilj pridobiti podatek o koraku gibanja. Za izvedbo tega koraka nisem uporabil algoritmov za prepoznavanje robov image flow in podobne tehnike, ampak je registracija gibanja odvisna od točnosti uporabnika. Korak gibanja je podatek, ki nam pove, za kolikšno razdaljo se je objekt premaknil, ali kolikšna razlika v pikslih je med prejšnjo in trenutno sliko, če vzamemo objekt, ki ga opazujemo kot izhodno točko. Za pridobitev podatkov o gibanju je potrebno določiti pot, ki ga je objekt opravil. Ker smo privzeli, da se bo naš objekt premikal linearno, je takšno gibanje dvodimenzionalno in je treba določiti le začetno in končno točko gibanja. Uporabnik izbere najprej začetno točko na prvi LR sliki iz množice, kjer pridobimo koordinate točke s klikom. Po izbiri prve točke uporabnik izbere še končno točko gibanja na zadnji sliki v LR množici slik imgs[]. Metodo za izračun koraka gibanja lahko zapišemo kot:

$$movex = (float)(endp.x - startp.x)/(float)(imgNum - 1),$$

kjer je movex korak gibanja za koordinato x, endp.x in startp.x sta končna in začetna točka. Na podoben način izračunamo movey. Razliko med končno in začetno točko delimo za število LR slik v imgs[] množici -1, ker začetno LR sliko ni potrebno premikati. Končni rezultat registracije gibanja sta spremenljivki movex in movey, ki pokažeta gibanje objekta za obe

koordinati. Če uporabnik pozna vrednost koraka za x in y os, lahko preskoči fazo registracije. Ob tem pa definira v nastavitvah movex in movey tako, da nista enaka 0.

Obnovitev slik

Dobljene podatke do sedaj uporabimo kot argumente za SR funkcijo, ki nas pripelje do končne ocene HR slike. Funkcijo SR sem definiral na sledeči način:

$$makeSuperResolution\left(\begin{matrix} Mat\ in[\ ], Matout, int\ imgNum, int\ scale, Point2d\ step, \\ int\ iterations, float\ alpha, float\ beta, float\ lambda, Size\ \text{ker} \end{matrix}\right);$$

kjer je in[] vhodna matrika LR slik, v našem primeru imgs[], &out je kazalec na ciljno matriko, step je točka (x,y), za katero velja, da x= movex in y = movey, ker določata velikost okna filtra.

Funkcijo makeSuperResolution lahko razdelimo na 2 dela: interpolacija in obnovitev slik. Poleg tega je metoda iterativna, torej je nujna ponovitev vsakega dela za določeno število iteracij (iterations). Kot začetno oceno HR slike vzamemo prvo sliko iz množice imgs[] in jo shranimo v (1D) matriko outVec. Takšna slika je linearno povečana, neobdelana s SR tehniko in zato so vidna popačenja na sliki, ki jih dobimo s preslikanjem LR elementov na HR mrežo. Za vsako opravljeno iteracijo metode se bo ocena HR slike približala idealni oceni HR slike. Koliko se približamo idealni oceni HR slike, je odvisno od napak v merjenju, metode SR in številu vhodnih LR slik.

Poravnava

Za vsako iteracijo moramo primerjati množico LR slik s trenutno oceno HR slike outVec. Primerjavo med slikami izvedemo na ta način, da za vsako LR sliko premaknemo trenutno oceno HR slike outVec na takšen način, da se outVec ujema z vsako LR sliko. Premik za vsako sliko definiramo tako:

$$foreach(x)\{foreach(y)\{Out[x + k * step.x, y + k * step.y]_k = Out[x, y]\}\}$$

$$0 < k < imgNum,$$

kjer je Out 2D matrika outVec, katero za vsako LR sliko na mestu k premaknemo globalno celotno matriko za razdaljo k*step pixlov. Funkcijo za premikanje slik sem definiral kot:

$$Mat\ moveImg(Mat\ img, int\ x, int\ y),$$

kjer je img vhodna matrika, za katero izvedemo globalni premik pixlov za vrednosti x in y, kjer je x premik po x osi in y je premik po y osi. Funkcija vrne kot rezultat matriko, ki vsebuje vrednosti o premaknjeni sliki. Slike, ki jih dobimo s premikanjem HR ocene, shranimo v polje estSet[], ki predstavlja množico premaknjenih ocen.

Primerjava slik

Naslednji korak je primerjava slik iz množice estSet[] z vhodnimi LR slikami. Primerjavo slik sem definiral z naslednjo funkcijo:

$$signDiff(\ Mat\&\ in1, Mat\&\ in2, Mat\&\ out),$$

kjer in1 in in2 kažeta na vhodne matrike, ki jih primerjata med sabo in out je kazalec na izhodno matriko. Matrike in njihove vrednosti primerjamo za vsak pixel za definirani način: (in1>in2):

$$(in1 > in2): return\ 1; (in1 < in2): return\ -1; (in1 = in2): return\ 0;$$

kjer glede na vrednosti in1, in2 dobimo normalizirani rezultat spremembe. Funkcija signDiff nam vrne matriko enake velikosti vhodnih matrik, ki vsebuje dobljene vrednosti primerjav med matrikami. Te hranimo v matriki matDiff.

Obnavljanje slik

Sliko obnovimo v treh korakih: z združitvijo, regularizacijo in poostrivijo. Trenutno HR oceno slike outVec najprej združimo z vsemi matDiff matrikami. Za združitev sem uporabil metodo:

$$addWeighted(Mat\&\ src1, double\ alpha, Mat\&\ src2, double\ beta, double\ gamma, Mat\&\ dst);$$

kjer je src1 vhodna matrika, ki ima težo v vsoti matrik enako alpha, src2 je druga vhodna matrika s težo v vsoti beta, gamma je skalar, dodan končni vsoti in dst je izhodna matrika, v katero se zapiše rezultat. Za naš primer združitve matrik imamo vrednosti:

$$addWeighted(outVec, alpha, matDiff_k, -beta, 0.0, outVec);$$

Ker z metodo dodajamo vhodni sliki oceni SR, matriko izračunanih napak (razlik) za vsako LR sliko in alpha predstavlja težo prve slike, beta pa težo razlik na sliki, je vrednost alpha enaka 1.0, saj hočemo, da originalna slika ostane enaka s spremembami na mestih, kjer so bile zaznane razlike. Kolikšen vpliv ima matrika razlik na originalno oceno SR, je definirano z beta, kjer večja vrednost pomeni večji vpliv na oceno SR. Vrednost gamma je 0.0, saj v večini primerov ne potrebujemo globalnega množenja vrednosti na sliki s skalarjem gamma.

Dobljeni rezultat regulariziramo z bilateralnim filtriranjem. Bilateralni filter služi odstranjevanju šuma na sliki, kjer upošteva in ohranja robove. Filter izračuna nove vrednosti pikslov glede na povprečje vrednosti sosednjih pikslov. Funkcijo za izračun regularizacijske matrike sem definiral kot:

$$bilateralReg(Mat\&\ inVec, Size\ ker, float\ alpha, Mat\&\ outVec, Size\ s);$$

kjer je inVec vhodni vektor, ker predstavlja velikost okna filtra, alpha je parameter prostorne distribucije, outVec je kazalec na matriko, v katero zapišemo dobljeni rezultat in s predstavlja velikost 2D matrike. Regularizacijska matrika je matrika zapisana na mestu outVec, kjer vrednosti elementov v matriki predstavljajo učinek bilateralnega filtriranja.

Dobljeno matriko odštejemo trenutni SR oceni z uporabo funkcije addWeighted, da odstranimo šum, dobljen po interpolaciji slik v SR metodi. Po dodani regularizacijski matriki sem uporabil unsharp mask, kjer z uporabo Gaussove zameglitve sestavimo masko iz končne SR ocene. Dobljeno masko primerjamo in odštejemo od SR ocene, kar odstrani del zameglitve na SR oceni.

Postopek ponavljamo za določeno število iteracij, kjer imamo po končani zadnji iteraciji najboljšo možno oceno HR slike za dane vhodne podatke in za izbrani algoritem SR obnovitve.

5 TESTNI PRIMERI

Za testiranje delovanja programa sem se odločil za različne primere, s katerimi sem preveril vpliv vhodnih podatkov in nastavitev na končni rezultat.

5.1 Delorean - model avtomobila

V prvem primeru sem zajel z uporabo spletne kamere množico 100-tih slik modela avtomobila, ki sem ga za vsako sliko premaknil za 1 mm. Kamere nisem premikal in edini premiki na sliki so premiki modela avtomobila. Vhodno število uporabljenih LR slik za ta primer je 16. Vrednost koraka na x osi je (28.0/15.0) in objekt se ni premikal po y osi, zato je korak na y osi enak 0. Povečava slik je določena z vrednostjo scale, ki je enaka 2 za ta in naslednje primere. Večja povečava ni uspela, saj je zapolnilo virtualni pomnilnik. Super resolucijo izvedemo s 180 iteracijami, kjer je alpha 0.6f, beta 0.2f in lambda 0.03f. Okno filtra ker je enako velikosti Size (7,7).

Slika 4: Rezultati na modelu avtomobila Delorean, levo LR slika, desno SR rezultat

Na slikah avtomobila lahko opazimo večjo točnost v podrobnostih. Opazna je izboljšava na kablih, ki ležijo nad kolesi, izboljšan pogled v notranjost avtomobila in boljša ohranjenost oblike. V zadnjem delu avtomobila opazimo tudi izboljšavo na kablih in opremi. Ker se je premikal le avtomobil, opazimo poslabšanje v ozadju, kjer so podrobnosti nerazpoznavne. Enako velja tudi za kolesa avtomobila, ki so se ob premikanju vrtela.

5.2 Ladja

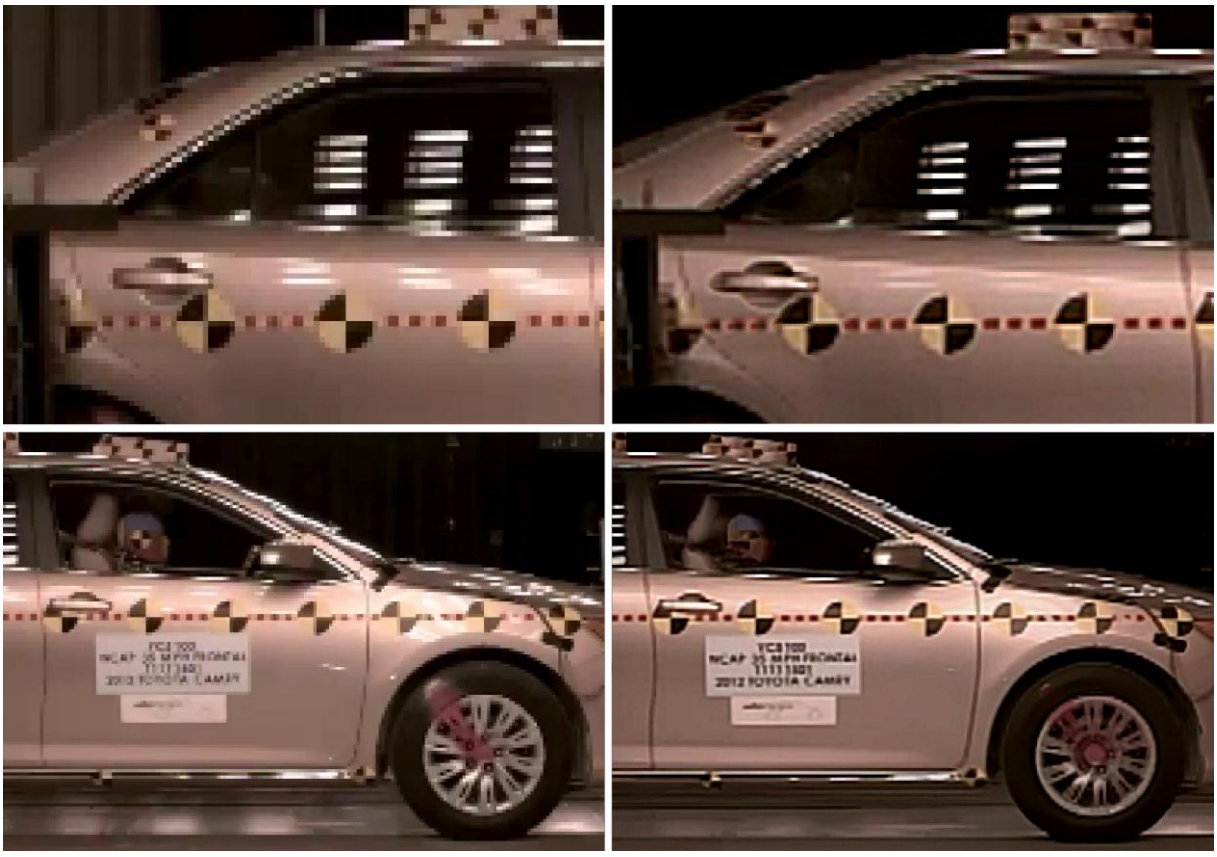
Za drugi primer sem postavil stacionarno kamero, ki je zajela posnetek ladij, prihajajočih k pomolu. Kamera je bila postavljena ob obali. Tako sem lahko dobil zaporedje slik, ki sem jih pomanjšal in nato poskusil obnoviti z metodo SR. Vhodno število uporabljenih LR slik za ta primer je bilo 50. Vrednost koraka na x osi je (220.0/49.0), vrednost koraka po y osi pa je (7.0/49.0). Zanimivost tega primera je linearni premik na x in y osi. Ladja se je premikala linearno na eno stran, vendar so zaradi valovanja pričakovane napake manjše za oceno gibanja. Več napak sem pričakoval na y osi. Super resolucijo sem izvedel s 180 iteracijami, kjer je alpha 0.6f, beta 0.05f in lambda 0.03f. Okno filtra ker je enako velikosti Size (7,7).

Slika 5: Rezultati za ladjo, levo LR slika, desno SR rezultat

Na slikah lahko opazimo izboljšave v zadnjem delu ladje, na okrasni črti na boku ladje, kjer LR slika vsebuje nepravilnosti in je črta videti ″razbita″. Spremembe vidimo na sedečih ljudeh, ki se niso veliko premikali, medtem ko pa sta se ostali dve osebi premikali po ladji in to takoj opazimo kot zameglitev, ki je nastala s premikom. V zadnjem delu ladje je videti več podrobnosti na jadru, kjer na SR obdelani sliki lahko opazimo, kako je bilo jadro zloženo. Na sprednjem delu ladje vidimo izboljšavo predvsem na številki ladje in boljše definirano črto ter jasno definirana okna. Človek se je premikal od desne proti levi in je tako na sliki videti njegovo zamegljeno premikanje. Jadrnica v ozadju se je obračala, vendar se ni premikala v isto smer kot ladja, zato je v SR obdelani sliki videti na njeni višini manjše napake v ozadju. Poleg jadrnice lahko opazimo v ozadju del kopnega, ki je po SR obdelavi videti razvlečeno.

5.3 Toyota

Zadnji primer sem vzel iz video posnetka ″crash″ testa avtomobila [11] LR slike. Za SR obnovitev sem uporabil zaporedje 50 takih slik. Pred trkom v steno se je avto premikal s konstantno hitrostjo 35 mph. Korak štejemo samo na x osi za vrednost(137.0/49.0). Ta primer, sicer podoben prvemu, ima na sliki več podrobnosti in uporabil sem večje število slik za SR obnovitev. Super resolucijo sem izvedel s 180 iteracijami, kjer je alpha 0.6f, beta 0.05f in lambda 0.03f. Okno filtra ker je enako velikosti Size (7,7).



Slika 6: Rezultat za ″crash″ test Toyote, levo LR slika, desno SR rezultat

Rezultati so pokazali veliko pridobitev na podrobnostih in natančnejše definirane oblike. Največja izboljšava po SR obnovitvi LR slik se je pokazala v berljivosti teksta na nalepkah na avtomobilu in na posebnih oznakah. Če pogledamo LR slike, lahko opazimo črtkano črto med črno-rumenimi krogi. V primeru LR slik ni gotovo, ali je črta sestavljena iz pik ali manjših črtic. Ko pogledamo SR obdelane slike, opazimo pravo obliko črte in spoznamo, da je sestavljena iz manjših črtic, ne pa iz pik. Videz voznika v avtomobilu se je popačil, saj se ni premikal enakomerno z avtomobilom. Zanimivost pri tem rezultatu je, da so se odsevi ohranili, čeprav se je avtomobil premikal. Enako velja za kolesa. Kljub premikanju po SR obnovitvi niso vidne večje napake razen pri delu kolesa, ki je obarvano z rdečo barvo. Tam je jasno opazen

efekt premika, saj ga ni bilo mogoče poravnati z našim linearnim gibanjem avtomobila. Ker je efekt premika na rdečem delu opazen in na ostalem delu kolesa ni opaznega premika, čeprav se je kolo premikalo enako z rdečim delom, sklepam, da gre za večji kontrast med kolesom in črnim ozadjem. Največjo težo ima prva slika v zaporedju, ostale slike pa dodajo svoje vrednosti. Te so nato regularizirane, zato so bile po vsej verjetnosti pobrisane. Enako si razlagam rezultat pri odsevih na avtomobilu.

6 ZAKLJUČEK

Tema moje diplomske naloge je bila prikaz enostavnega algoritma za SR obnavljanje slik. Prikazal sem osnovni opazovalni model super resolucije, kjer je predstavljeni problem definiran kot inverzni, saj je potrebno izvesti obratne operacije funkcij, ki so popačile sliko v času njenega zajemanja. Za izvedbo super resolucije sem potreboval več pogledov na isto sceno in iz vseh možnih sistemov sem se odločil za sistem z nepremično kamero, kjer se premika objekt, ki sem ga nato obnavljal z algoritmom SR.

Za preizkus iterativnega SR algoritma sem predvideval, da je gibanje linearno in med popačenji upošteval, da je šum beli in zameglitev na slikah Gaussova zameglitev.

Iz rezultatov sem ugotovil, da je SR algoritem občutljiv na netočnosti pri meritvah, kjer je v prvem primeru bilo gibanje težje preslikati na HR mrežo in je zaradi netočnosti prišlo do popačenj na delih slike. Kadar je linearno gibanje zagotovljeno z enakomerno hitrostjo, sem dobil veliko boljše rezultate, kjer je učinek super resolucije bil jasno viden pri napisu na avtomobilu.

Zaradi računske zahtevnosti algoritma sem bil omejen z maksimalno povečavo slike po x in y osi. Z nastavitvijo prevelike povečave se virtualni pomnilnik zasede in ni mogoče nadaljevati z izvajanjem SR. Kot izboljšave za metodo SR je treba zagotoviti večjo stabilnost SR obnovitev, če obstajajo napake v meritvah in zmanjševanje računske zahtevnosti algoritma.

Postopek SR je uporaben glede na rezultate za razpoznavanje podrobnosti v slikah, kjer so le-te zarisane nejasno. Na področju medicine in kriminalistike že uporabljajo super resolucijo za zmanjšanje napak pri odločitvah. Predvidevam, da bodo SR metode koristile tudi na področjih mobilnih aplikacij, in to za izboljšavo kvalitete zajetih slik, kjer bo pomembna hitrost izvajanja in nizka poraba virov. Super resolucija omogoča podrobnejši pogled na slike, ki jim primanjkuje podrobnosti.

7 LITERATURA IN VIRI

[1] Sina Farsiu, M. Dirk Robinson, Michel Elad, Peyman Milanfar (2004), Fast and Robust Multiframe Super Resolution, IEEE Transactions on image processing vol.13, no.10

[2] Michael elad, Yacov Hel-Or (1998), *A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space-Invariant Blur*, Israel, HP Laboratories

[3] Sung Cheol Park, Min Kyu Park, Moon Gi Kang (2003), *Super-Resolution Image Reconstruction: A Technical Overview*, IEEE Signal Processing Magazine 1053-5888/03

[4] Sean Borman, Robert L. Stevenson, Super-Resolution from Image Sequences- A Review, University of Notre Dame

[5] Sina Farsiu, Dirk Robinson, Michael Elad, Peyman Milanfar (2004), Advances and Challenges in Super-Resolution, Santa Cruz, University of California

[6] Dirk Robinson, Peyman Milanfar (2006), *Statistical Performance Analysis of Super-Resolution*, IEEE Transactions on image processingm vol. 15, no.6

[7] Gilles Rochefort, Frédéric Champagnat (2009), Guy Le Bersnerais, Jean-François Giovanelli, *An Improved Observation Model for Super-Resolution under Affine Motion*, arXiv:0908.3250v1

[8] Hong Chang, Dit-Yan Yeung, Yimin Xiong, *Super-Resolution Trough Neighbor Embedding*, Hong Kong University of Science and Technology

[9] B. Bascle, A. Blake, A. Zisserman, *Motion Deblurring and Super-resolution from an Image Sequence*, University of Oxford

[10] He He, Wan-Chi Siu, *Single Image Super Resolution using Gaussian Process Regression*, Hong Kong Polytechnic University

[11] CrashNet1, *Toyota Camry | 2012 | Frontal Crash Test | NTHSA High Speed Camera | CrashNet1*, http://www.youtube.com/watch?v=ZhwiHdO5OBc