

Implementacija algoritma Posplošena interpolacija z omejitvami

Tekst vsebuje kratek povzetek opravljenega dela in pridobljenih rezultatov ter opis nerešenih problemov in vprašanj glede implementacije.

SPLOŠNO

Izbran algoritem za povečavo slik sestavljajo trije modeli. Sliko najprej povečamo z linearno interpolacijo, nato pa s pomočjo modelov popravimo vizualne napake in jo približamo videzu realnega sveta. Model senzorja skrbi, da povečana slika pretirano ne odstopa od originalne slike, model ostrenja in glajenja pa skrbita za ostrino robov. Postopek je iterativen.

1. MODEL SENZORJA

Model senzorja skrbi, da je popravljena povečana slika konsistentna z originalno sliko, da ostrenje in glajenje robov ne popači slike. To doseže tako, da primerja povprečena območja visokoresolucijske (VR) slike in piske originalne nizkoresolucijske slike (NR). Skladno z razliko model popravlja vrednosti visokoresolucijskih pikselov. Povedano z drugimi besedami, vsak piksel osnovne slike se mora ujemati z izbranim območjem visokoresolucijske slike. Ime modela izhaja iz posnemanja proces zajemanja slike, kjer senzor podobno povpreči okolico in informacijo zapiše v posamezen piksel.

Rezultati



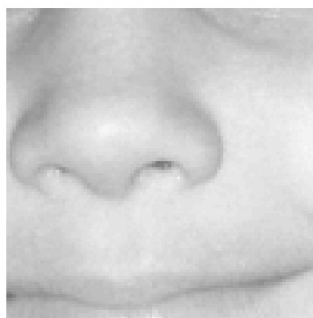
(a) originalna slika Test.



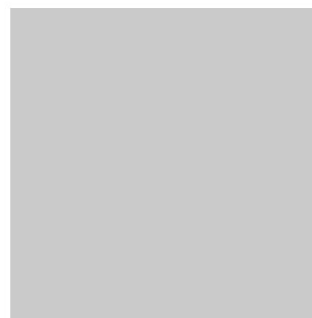
(b) Povečana slika (2x2)
z dodanim šumom.



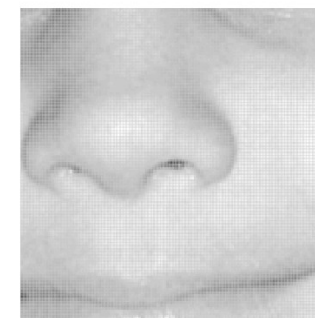
(c) Popravljen slika po 50
ponovitvah.



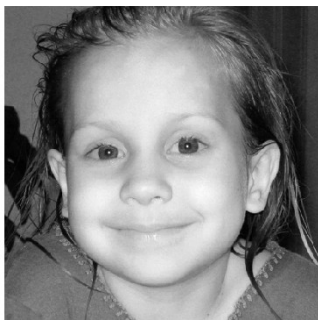
(a) Originalna slika, del slike
Rachel.



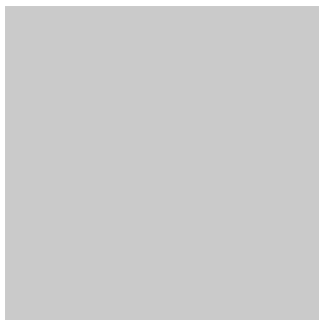
(b) Povečana slika oz.
prazno platno za namen
testiranja.



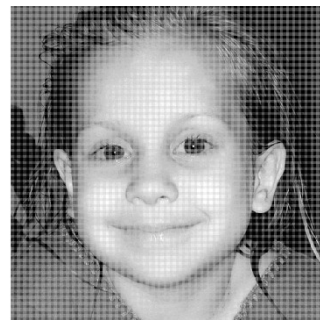
(c) Popravljen slika oz.
rekonstruirana slika po 10
ponovitvah.



(a) Originalna slika, del slike Rachel.



(b) Povečana slika oz. prazno platno za namen testiranja.



(c) Popravljen oz. rekonstruirana slika po 10 Ponovitvah.

Ime slike	Dimenzija	Število ponovitev	Čas
Test	6 x 5	50	0.6 s
Rachel	100 x 100	10	30 s
Rachel	400 x 400	10	8 min
Rachel	512 x 512	10	12 min

Tabela 1: Časi obdelave, število ponovitev in dimenzije v številu pikslov.

Diskusija: Model opravlja svojo funkcijo in povečano sliko približuje originalni sliki. Da bi sliko zadovoljivo rekonstruirali, bi potrebovali več iteracij. Ker imamo potrebo zgolj po ohranjanju podobnosti, bomo natančnejše merite izdelovali po združitvi modelov.

2. MODEL GLAJENJA

Model glajenja si lahko predstavljamo kot proces krajšanja krivulj. Krivuljo v našem primeru predstavljajo robovi med plastnicami pikslov enake ali podobne barve. Cilj algoritma je glajenje cikcaka (jaggies), ki je pogosto stranski produkt interpolacije.

Sliko gladimo tako, da najprej izračunamo ukrivljenost plastnic K . Bolj kot so ukrivljene, bolj jih želimo poravnati. Upoštevamo tudi magnitudo gradienta (razliko v vrednosti pikslov). Visok gradient pomeni velike spremembe v vrednostih pikslov, zato so tudi popravki večji in obratno. Popravljamo zgolj vrednosti pikslov in s tem premikamo plastnice, pozicij plastnic ne računamo.

Formule

$$I_t = -\kappa \|\nabla I\| ; \quad \kappa = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{\frac{3}{2}}}$$

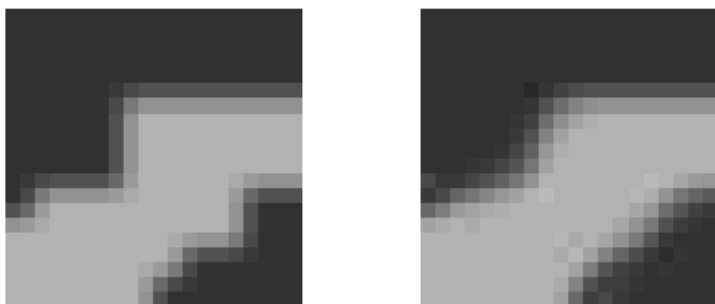
I_t ... Korekcijska vrednost piksla t

κ ... Ukrivljenost plastnice

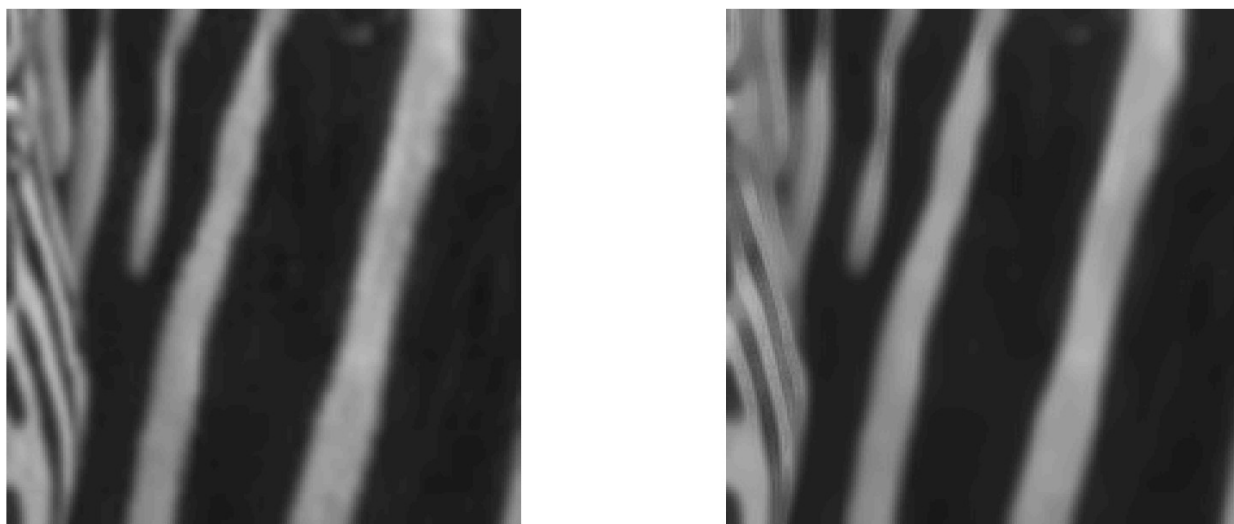
$\|\nabla I\|$... Magnituda gradienta piksla t

I_x ... Odvod v pikslu t po x (in ujemajoče za vse ostale odvode)

Rezultati



(a) Primer glajenja na preprosti sliki. Levo original in desno slika po dveh ponovitvah glajenja.



(b) Originalna slika Zebre (levo) in slika Zebre po 6. ponovitvah glajenja (desno).

Ime slike	Dimenzija	Čas(s)	Število iteracij
Zebra	100 x 100	3	2
Zebra	464 x 329	45	2
Rachel	512 x 512	80	2

Tabela 2: Dimenzije v številu pikslov, čas obdelave, število ponovitev glajenja.

Glajenje je ob pravilnem številu ponovitev lepo izvedeno. Pri sliki zebrinih črt lahko vidimo, da je na širokih barvnih pasovih slika lepo glajena, pri ožjih pasovih (ob levem robu) pa pride do pretiranega glajenja. Za ožje pasove sta primernejši le 2 ponovitvi glajenja.

Včasih se pojavijo šumi (presvetljeni ali pretemnjeni kvadrati na gladkih površinah). Predpostavka je, da nastanejo zaradi pretiranega glajenja. Možne rešitve so: dodan senzor model, dodan kriterij za prekomerno glajenje ali pa izbira boljšega postopka/ formule za glajenje. Prvo opcijo bom stestirala ob združevanju modelov, kriterij za prekomerno glajenje pa je opisan spodaj, z njim imam določene probleme.

2. 1. STOP KRITERIJ ZA MODEL GLAJENJA

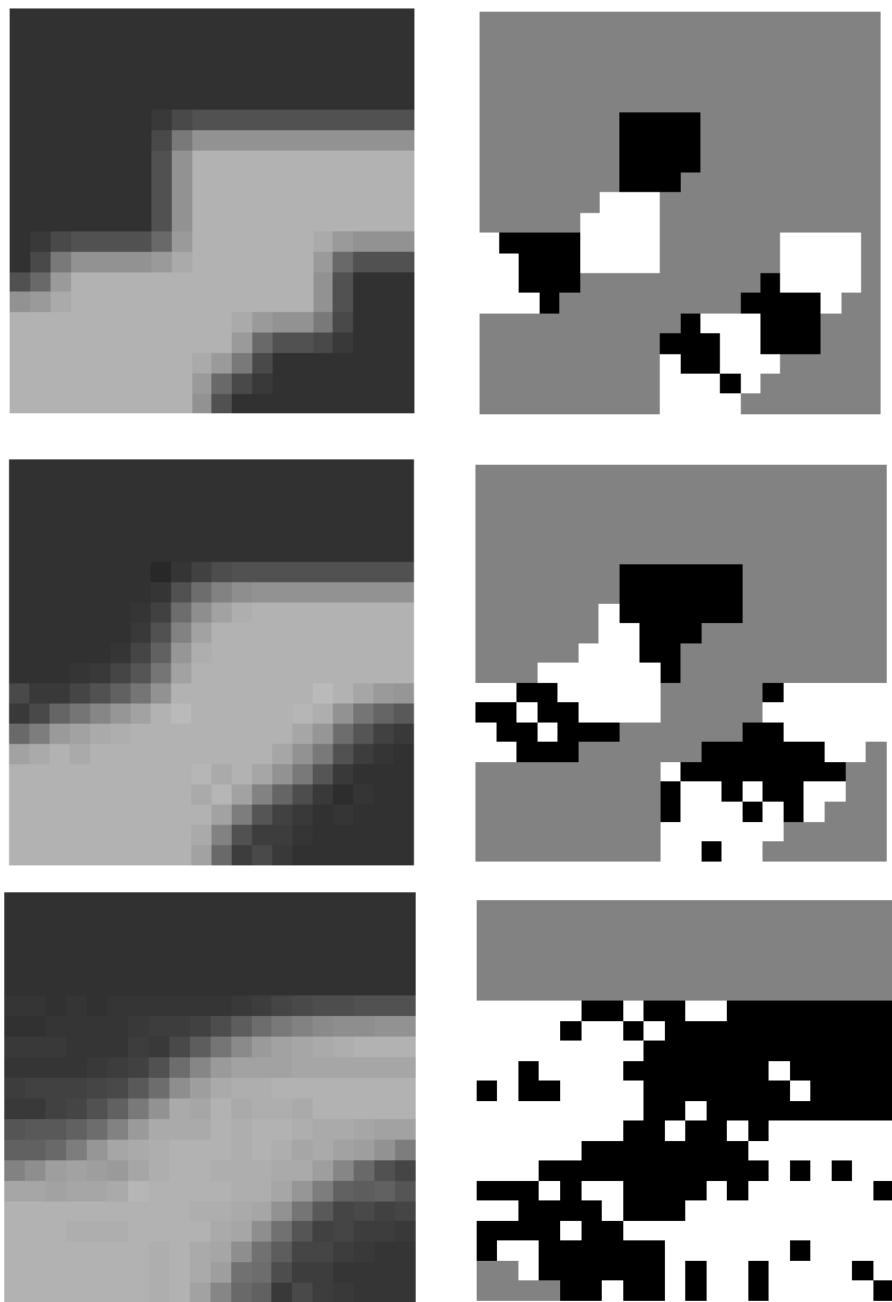
V članku za stop kriterij uporabljamo masko pregibov (inflection mask). Masko počasi označujemo in tako zaklepamo piksele. Ko določen piksel na maski označimo, ga ne smemo več gladiti. Ko je celotna maska označena, se glajenje ustavi. Posamezne piksele označimo, ko so plastnice (contours) v njegovi okolici ali konveksne ali konkavne – ko v plastnicah ni pregibov. Če bi glajenje po tem še nadaljevali, bi iz glajenja plastnice prešli v krajšanje.

V algoritmu okolica velja za konveksno oz. konkavno, ko je predznak “of the isophote flow” (interpretirano kot predznak ukrivljenosti plasnitc) v vsej okolici enak. Za vsak piksel sproti preverimo okolico. Ker celotna slika v nekem trenutku ne more biti konveksna, piksela ne ne spreminjamo več, ko je njegova okolica prvič konveksna. Za okolico vzamemo nekoliko večje okno od faktorja povečave, da zajamemo celotno “enoto cikcaka”.

Formula po kateri se računa konveksnost oz. konkavnost je sledeča:

$$\kappa = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{\frac{3}{2}}}$$

Rezultati



(a) V levem stolpcu so prikazane slike, v desnem pa izračunana konveksnost oz. konkavnost posameznega piksla.

Diskusija

Opisan postopek ima v praksi več problemov:

1. Točke pregibov se skozi proces glajenja ohranjajo, kar je tudi želeno. Želimo ohranjati obliko slike in ne izravnati vseh robov v ravne črte. Ker se pregibi ohranjajo, traja veliko časa, da postane okolica pikslov v pregibnih delih enako predznačena. Če pikslov ne zaklepamo in pustimo procesu da teče, je rezultat sicer pokrita maska, ampak popolnoma izmaličena slika. Ta proces je prikazan na zgornji sliki.
2. Če piksle ob doseženem pogoju (enako predznačena okolica) zaklenemo, potem pregibi ostanejo in nikoli ne pokrijemo vseh pikslov ma maski, zato se proces ostrenja ne ustavi.

Alternativa

Kot alternativni pogoj za ustavljanje bi lahko izbrali količino ukrivljenosti na posameznem pikslu. Okolica, ki jo preverjamo, mora biti za ena večja od faktorja povečave, da se lahko znebimo cikcaka. To bi lahko določili s številom pikslov, ki jih upoštevamo pri računanju odvodov. Za dvakratno povečavo in manj zadošča osnoven način računanja odvodov s pomočjo sosednjih pikslov, s tem zajamemo ukrivljenost na okolici 3×3 . Za večje povečave bi uporabili druge metode. Pogoj za zaklep piksla bi bil določen prag ukrivljenosti blizu nič. Kar bi pomenilo, da je rob lokalno skoraj raven. To nam ustreza, saj smo se znebili jaggies. Problem lahko nastane pri ostrih kotih v sliki.

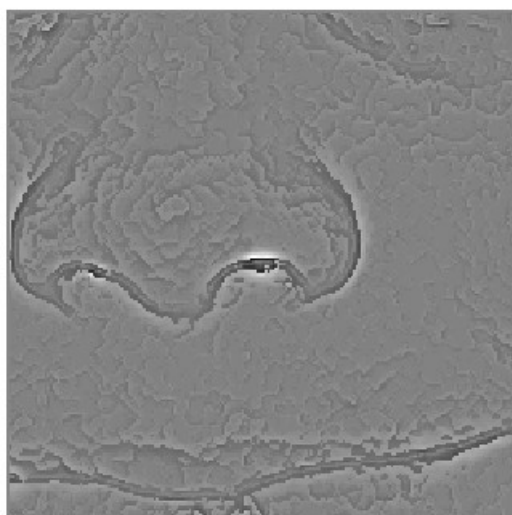
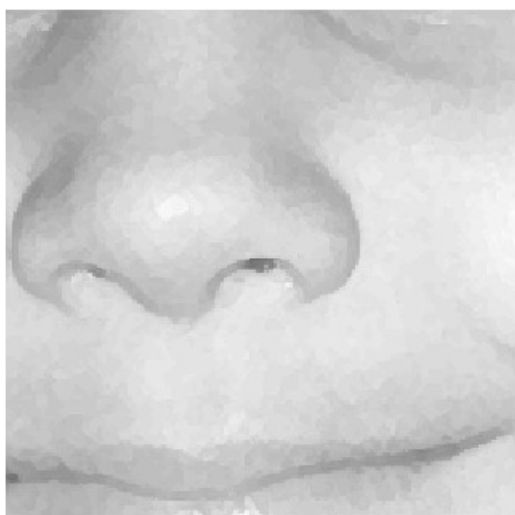
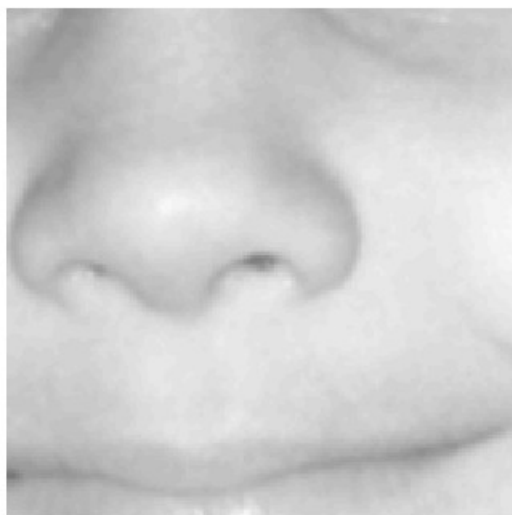
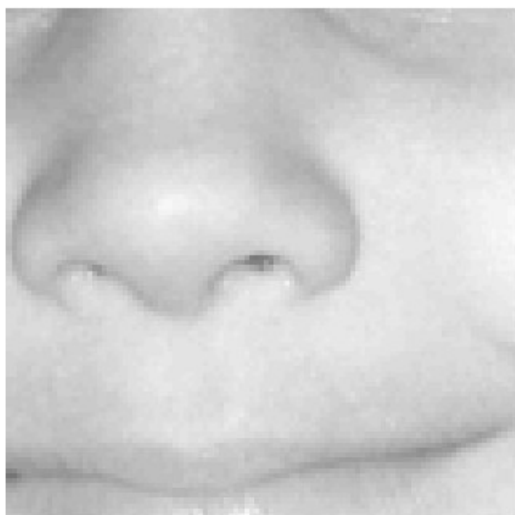
Predstavljeno idejo moram stestirati. Preveriti moram ali je in koliko je ukrivljenost odvisna od vrednosti pikslov in razlik med njimi. In koliko in kaj bi bilo glede na to smiselno vzeti za prag.

3. MODEL OSTRENJA

Ostrenje poteka po principu postopnega oddaljevanja vrednosti pikslov (nudge flow) od vrednosti na drugi strani roba. Na primer, če gre rob iz temne v svetlo in se piksel najaha na svetlejšem delu, ga bo model posvetlil, če pa je na temnejem delu roba, ga bo potemnil. Posledično se vrednosti na obeh straneh roba vedno bolj razlikujejo in robovi postajajo jasnejši.

Ime	Dimenzije	Čas (s)
Rachel	100 x 100	2
Rachel	512 x 512	35

Tabela 3: Dimenzije v številu pikslov in čas obdelave za eno ponovitev ostrenja.



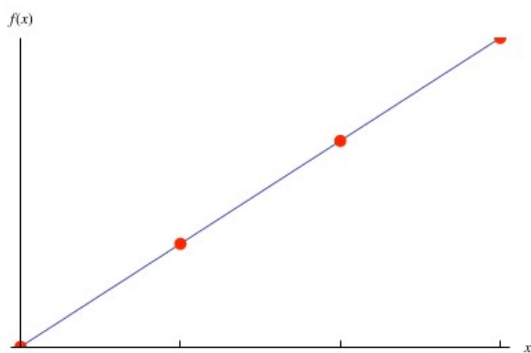
Levo zgoraj originalna slika, desno zgoraj bilinearna povečava 2x2, **levo spodaj izostrena slika**, desno spodaj vrednosti popravkov po formuli postopnega oddaljevanja vrednosti.

Implementacija ostrenja deluje pravilno, uspeli smo ponoviti rezultate iz prejšnjih raziskav. Na ostreni sliki se poznajo manjše okolice, ki so enotne barve, kar pomeni, da imamo pretirano ostrenje. Zato v drugem delu dodamo stop kriterij.

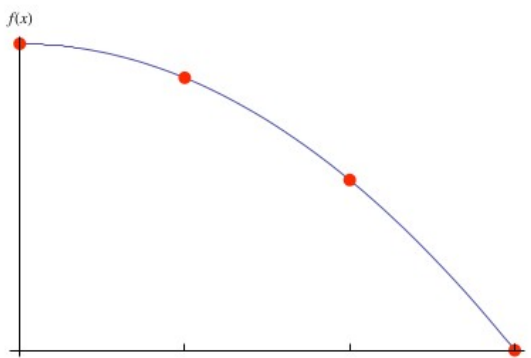
3. 2. STOP KRITERIJ ZA MODEL OSTRENJA

Stop kriterij s pomočjo najbližjih NR pikslov izračuna največji dovojen naklon med izbranim in sosednjimi VR piksli. S stop kriterijem imam probleme. V prvem delu je opisano delovanje algoritma v drugem pa problematika. Najprej so za občutek opisani primeri robov s katerimi se ukvarjamo.

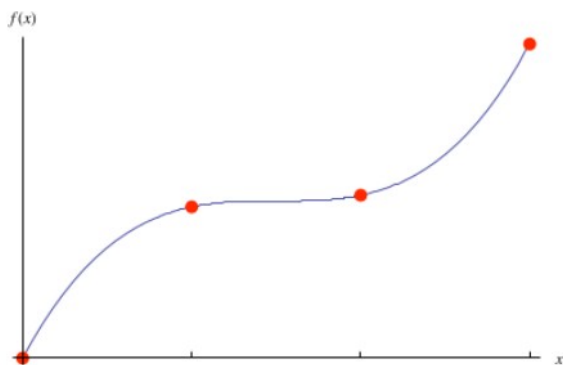
1) Primeri, ko ostrenje ni dovoljeno.



(a) Example of collinear low-resolution points. No sharpening allowed.

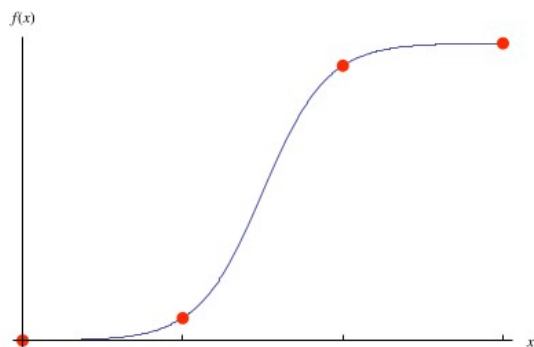


(a.2) Ukrivljenost se počasi spreminja, nimamo točke preloma, zato imamo blago, minimalno glajenje.

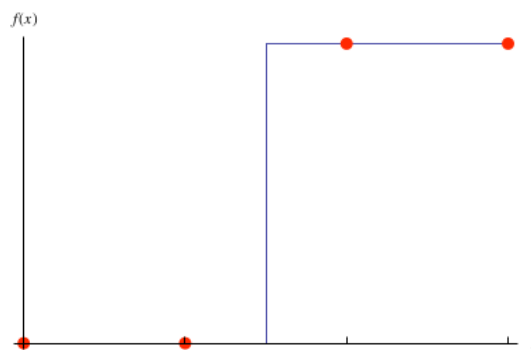


(d) Obstaja točka preloma, ampak ne na pravem mestu.

2) Primeri, ko je glajenje dovoljeno.



(c) Example of a soft edge. Only partial sharpening allowed.



(b) Example of a step edge in the low-resolution image. Maximal sharpening allowed.

Algoritem za izbran piksel povečane slike (VR) najprej izračuna najbližja piksla iz originalne slike (NR) v smeri gradienta VR slike. Glede na razliko njunih vrednosti in ukrivljenosti izračunamo največji dovoljen naklon (razliko vrednosti) za izbran HR piksel in njegove sosede. Naklon izračunamo po formuli:

$$M = \left(\frac{|I_{i+1} - I_i|}{X} \right) \left(1 + (x - 1) \left[\frac{|C_{i+1} - C_i|}{2 |I_{i+1} - I_i|} \right]^\alpha \right)$$

I ... Vrednost sosednjega NR piksla (v smeri gradienta HR piksla).

C ... Ukrivljenost plastnice pri najbližjem NR pikslu (v smeri gradienta HR piksla).

X (in x) ... Faktor povečanja.

α ... Moč ostrenja.

M1 ... Poimenovanje za prvi oklepaj.

M2 ... Poimenovanje za drugi oklepaj.

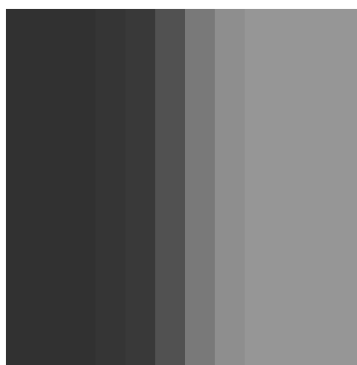
V osnovi ločimo primer (a), ko so vrednosti pislov kolinearne in primer (b), ki predstavlja stopničasti rob:

(a) Primer kolinearnih pikslov in pikslov brez prevoja. Razlika ukrivljenosti C je blizu 0, ulomek ima vrednost okoli nič in posledično se oklepaj M2 izenači z 1. Ko poračunamo ostane le prvi oklepaj in tako dobimo najblažje možno ostrenje – največji dovoljen naklon je razlika med sosednjima LR pikslova deljeno s faktorjem povečave.

(b) Primer stopničastega roba. V teoriji naj bi bila vrednosti ulomka v drugem oklepaju (M2) enaka 1. Ko bi poračunali, bi dobili največje možno glajenje - naklon z razliko vrednosti LR pikslov. V praksi se tukaj zalomi, ker so rezultati ulomka iz M2 neomejeni (pogosto delimo z vrednostmi blizu 0).

Op: Formula M pokriva tudi umesne primere blažnjih robov, ki so bili predstavljeni s slikama (c), in posebne primere, kot je na sliki (d).

Vhodni podatki



a) Vhodni podatek: slika 6 x 6 pikslov.

Prikazuje omiljen stopničasti rob. (im_org)

b) Dvakrat povečana slika z linearno interpolacijo.

Je osnova na kateri delamo ostrenje. (im_mag)

Razmislek

$$M = \left(\frac{|I_{i+1} - I_i|}{X} \right) \left(1 + (x - 1) \left[\frac{|C_{i+1} - C_i|}{2 |I_{i+1} - I_i|} \right]^\alpha \right)$$

Velja:

$$I \in [0, 255], \quad \Delta I \in [0, 255],$$

$$C \in [10^{-8}, 10^8], \quad \Delta C \in [0, 10^{16}].$$

Ostrenje ni dobrodošlo, ko je ukrivljenost plastnice ΔC nizka ali razlika med piksli ΔI nizka, zato bi nas namesto ulomka M2 zanimala formula oblike $f(\Delta C) * g(\Delta I)$.

Maksimalno ostrenje želimo, ko je ukrivljenost C velika. Če je v tem primeru ΔI blizu 0, nam formula oblike $f(\Delta C) * g(\Delta I)$ ne ustreza, saj želimo poračunano vrednost formule enako 1.

Nadaljne dileme:

Zanima nas odvisnot ukrivljenosti C od ΔI . Za različne stopničaste robove, dobimo različne ukrivljenosti C , pri vseh pa bi radi imeli največje možno ostrenje.

Stopnica ΔI	Ukrivljenost C
0 - 10	62.5
0 - 50	7812.5
0 - 100	62500
0 - 150	$2.1 * 10^5$
0 - 200	$5 * 10^5$
0 - 250	$9.8 * 10^5$

Tabela 4: Vrednosti C in ulomka M2 pri različnih ΔI za stopničasti rob.

Stopnica ΔI	Ukrivljenost C
50 - 100	7812.5
50 - 150	62500
50 - 200	$2.1 * 10^5$
50 - 250	$5 * 10^5$

Tabela 5: Vrednosti ΔC in ulomka M2 pri različnih ΔI za stopničasti rob.

V nadaljevanju je potrebno podrobno premisliti problem, konstruirati primerno formulo in stestirati alternativno rešitev.

4. SLOVAR STROKOVNIH IZRAZOV

Generalized constrained interpolation ... Posplošena interpolacija z omejitvami

Nudge flow ... Postopno oddaljevanje vrednosti